

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS FÍSICAS

Departamento de Arquitectura de Computadores y Automática



**FUSIÓN MULTISENSORIAL DE MEDIDAS
TEMPORALMENTE DESORDENADAS.
APLICACIÓN A ROBOTS AUTÓNOMOS
MÓVILES.**

**MEMORIA PRESENTADA PARA OPTAR AL GRADO DE
DOCTOR POR**

Eva Besada Portas

Bajo la dirección de los Doctores:

Jesús Manuel de la Cruz García
José Antonio López Orozco

Madrid, 2007



UNIVERSIDAD COMPLUTENSE DE MADRID

Departamento de Arquitectura de Computadores y Automática

Madrid, España

**Fusión multisensorial de medidas
temporalmente desordenadas.
Aplicación a robots autónomos
móviles.**

Tesis Doctoral

Dña. Eva Besada Portas

Madrid 2004



UNIVERSIDAD COMPLUTENSE DE MADRID

**Departamento de Arquitectura de Computadores y Automática
Madrid, España**

**Fusión multisensorial de medidas
temporalmente desordenadas.
Aplicación a robots autónomos
móviles.**

**Memoria para optar al grado de Doctor
presentada por**

Dña. Eva Besada Portas

Directores:

Dr. Jesús Manuel de la Cruz García

Dr. José Antonio López Orozco

A mis abuelas

Agradecimientos

Gracias a mis padres, José Luis y Lidia, por haber luchado toda su vida para darnos un hogar en el que desarrollarnos como personas y haber inculcado nuestro amor por la ciencia (aunque alguno vaya para artista). Desde pequeños habéis fomentado nuestras ganas de saber y despertado nuestra afición por el estudio. Muchas gracias por todo.

Gracias a mis hermanos por estar ahí, por hacer de hermanos durante tantos años, y por haberos leído esta tesis. A Juan Alberto, el mayor, por haber sido mi profesor particular desde que éramos pequeños y por ser el primero de los tres en encontrar un camino propio. A José Luis, el pequeño, por hacerme reír tanto, y por regalarme ideas originales, a veces disparatadas y siempre útiles. Tú encontraste un nombre apropiado a mi robot.

Gracias a mis “tíos”, Sara y Miguel, por estar cuatro pisos más arriba siempre que os hemos necesitado, aun cuando todos nos hemos mudado. Gracias a mis “parrinos”, Juan Enrique y Carmen, por contagiarnos con su sentido del humor, alegría y entusiasmo por la vida, y por el chocolate con churros cada año en San Ginés tras poner monigotes en la plaza Mayor. A todos vosotros, gracias por ser nuestra familia en Madrid.

Gracias, en fin, al resto de mi familia, por todo lo que me habéis dado. Además, de la familia que a uno le viene dada, extendiendo mi agradecimiento a la familia que he ido encontrando, sobre todo, cuando estaba lejos de España. Entre todos, habéis conseguido que los veranos y viajes hayan sido inolvidables.

Gracias a mis directores de tesis, los doctores Jesús Manuel de la Cruz García y José Antonio López Orozco, por haberme guiado a lo largo de este trabajo de investigación, al proporcionarme ideas y herramientas útiles para resolver los problemas que he ido encontrando, y por animarme cuando no veía la salida de los mismos. Sin vosotros, esta tesis no sería lo que es.

Gracias también a las demás personas con las que he trabajado durante todos estos años por hacer que éstos parezcan más cortos. A Bonifacio de Andrés por apostar por la gente joven y estar siempre dispuesto cuando se le necesita. A José María Girón y a los profesores de la Universidad de Castilla la Mancha Jorge Sanz, Eduardo Domínguez, Juan Pablo Rozas y José Flores por

orientarme, con sus sugerencias, durante el diseño del robot; y a Fernando Ginés y Carlos Roa, por ayudarme a construirlo. A Gonzalo Pajares por dirigir los proyectos que hacen que el robot pueda ver. A Óscar Rodríguez, por haber diseñado EdROOM. A Segundo Esteban, José Luis Risco, Joaquín Recas y Alberto Herrán por compartir tantas horas de investigación.

Gracias también al profesor Alex Gammernan por haberme acogido durante seis meses en el Royal Holloway de la Universidad de Londres, adentrarme en las técnicas de aprendizaje, y continuar preocupándose porque esta tesis llegase a buen fin.

Gracias finalmente a todos mis amigos, que habéis estado pendientes de mí a lo largo del desarrollo de este trabajo de investigación, y habéis entendido mis ausencias.

Índice

INTRODUCCIÓN.....	1
OBJETIVOS	3
<i>a. Desarrollo de métodos de fusión secuencial para la asimilación asíncrona de medidas retardadas y temporalmente desordenadas</i>	<i>3</i>
<i>b. Diseño e implementación de un sistema unificado de integración y fusión multisensorial con medidas temporalmente desordenadas para robots autónomos móviles</i>	<i>4</i>
ORGANIZACIÓN DE LA MEMORIA DEL TRABAJO REALIZADO	5
<i>a. Resumen del contenido de los capítulos.....</i>	<i>6</i>
<i>b. Resumen del contenido de los apéndices</i>	<i>8</i>
 CAPITULO I	
 FUSIÓN MULTISENSORIAL Y REDES BAYESIANAS.....	9
C1.1. INTEGRACIÓN Y FUSIÓN MULTISENSORIAL	10
<i>C1.1.1. Fusión Multisensorial</i>	<i>11</i>
▪ Clasificación de los procesos de fusión	12
Nivel funcional: tratamiento y uso que se hace de la información	12
Nivel de representación: tipo de información que combinan.....	13
Nivel algorítmico: técnicas utilizadas para fusionar la información.....	15
Nivel de implementación: arquitectura de control del sistema de fusión.....	16
▪ El problema de la recepción de medidas asíncronas, retardadas y temporalmente desordenadas en el sistema de fusión.....	17
▪ Aplicaciones de fusión	18
Fusión en robots autónomos móviles.....	19
C1.2. REDES BAYESIANAS	21
<i>C1.2.1. Funcionamiento de las Redes Bayesianas</i>	<i>22</i>
▪ Modelado gráfico	22

▪ Relaciones probabilísticas	23
▪ Algoritmos de inferencia probabilística	24
Algoritmos de inferencia exacta	26
Algoritmos de inferencia aproximada	27
<i>C1.2.2. Redes Bayesianas y Fusión</i>	<i>29</i>
▪ Algunos ejemplos del uso de redes bayesianas como técnicas de fusión.....	30

CAPITULO II

REDES BAYESIANAS PARA MODELAR SISTEMAS LINEALES DINÁMICOS CON SENSORES ASÍNCRONOS Y MEDIDAS TEMPORALMENTE RETARDADAS..... 33

C2.1. INFERENCIA EXACTA EN REDES BAYESIANAS CON VARIABLES CONTINUAS GAUSIANAS	35
<i>C2.1.1. Inferencia sobre árboles de unión para redes bayesianas con variables vectoriales continuas y distribuciones gaussianas multivariantes.....</i>	<i>36</i>
C2.2. REDES BAYESIANAS CONTINUAS Y FUSIÓN MULTISENSORIAL, ASÍNCRONA Y TEMPORALMENTE DESORDENADA	42
<i>C2.2.1. Redes Bayesianas equivalentes para sistemas lineales de control</i>	<i>44</i>
▪ Incorporación de la señal de control como evidencia en el potencial (C2-8)	49
▪ Marginalización de la señal de control del potencial (C2-8).....	49
<i>C2.2.2. Inferencia exacta y fusión sensorial asíncrona y/o temporalmente retardada sobre la red bayesiana que modela un sistema lineal de control con múltiples sensores</i>	<i>51</i>
▪ Estimación del estado del sistema para todos los instantes mientras no existe ninguna evidencia sensorial	55
▪ Incorporación de la evidencia sensorial y actualización del estado del sistema para todos los instantes de muestreo	58
▪ Generalización del algoritmo de inferencia del ejemplo para el caso en el que las medidas llegan al algoritmo de forma temporalmente desordenada	63
<i>C2.2.3. Algoritmo de inferencia exacta con una etapa de validación de medidas</i>	<i>65</i>
▪ Modelado de la etapa de validación como parte de la red bayesiana.....	66
▪ Tests de validación basados en la distancia entre el valor estimado y real de la medida	71
▪ Tests de validación basados en la distancia entre el valor estimado y real de la medida para un sistema con medidas asíncronas y temporalmente desordenadas.	76

CAPITULO III

**REDES BAYESIANAS DINÁMICAS PARA MODELAR SISTEMAS DE CONTROL
CON SENSORES ASÍNCRONOS Y MEDIDAS TEMPORALMENTE RETARDADAS..... 81**

C3.1. REDES BAYESIANAS DINÁMICAS PARA MODELAR EL COMPORTAMIENTO DE UN SISTEMA LINEAL CON SENSORES ASÍNCRONOS CON MEDIDAS TEMPORALMENTE DESORDENADAS	83
C3.2. FILTROS DE INFORMACIÓN PARA SISTEMAS DE CONTROL CON SENSORES ASÍNCRONOS CUYAS MEDIDAS PUEDEN ESTAR DISPONIBLES DE FORMA TEMPORALMENTE DESORDENADA	89
C3.2.1. <i>Filtro de Información para sistemas de control lineales con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada.....</i>	91
C3.2.2. <i>Filtro de Información Mixto para sistemas de control no lineales con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada.....</i>	98
▪ No linealidad en la función de transición: $f(x,u,t)$	99
▪ No linealidad en las funciones de medida $h_s(x,t)$	100
▪ Filtro de Información Mixto para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada.....	102
C3.2.3. <i>Características relevantes del filtro de Información Mixto para estimar el estado de un sistema de control con medidas temporalmente desordenadas y análisis comparativo con otros algoritmos de fusión multisensorial similares</i>	107
▪ Otros Métodos para la asimilación asíncrona de medidas temporalmente desordenadas	110
Métodos basados en el filtro de Kalman y la retrodicción del estado y su covarianza (sistemas lineales).....	110
Métodos basados en el filtro de Kalman y el precálculo de un conjunto de valores que permita la asimilación de la medida tan pronto como ésta se encuentre disponible (sistemas lineales).....	111
Método basado en el filtro de Información y el almacenamiento mínimo de variables (sistemas lineales).....	113
Métodos basados en el filtro de Kalman y el almacenamiento mínimo de variables (sistemas lineales).....	115
Métodos basados en el filtro de Kalman y la reformulación del problema por medio del uso de una variable de estado aumentada (sistemas lineales)	115
Métodos basados en el filtro de Partículas (sistemas no lineales).....	116
▪ Comparación conjunta de los algoritmos C3-2, C3-3 y los algoritmos asociados en las subsecciones anteriores.....	118

CAPITULO IV

ANÁLISIS DEL FUNCIONAMIENTO DE LOS ALGORITMOS DE FILTRADO

PROPUESTOS..... 121

C4.1. ANÁLISIS DEL FUNCIONAMIENTO DE LOS ALGORITMOS PARA SISTEMAS CON MODELOS

LINEALES 123

C4.1.1. Descripción general del sistema y de las pruebas realizadas..... 124

- Modelo del sistema y de los sensores..... 124
- Datos experimentales básicos..... 125
- Descripción de las Pruebas Experimentales..... 127
 - Grupo Experimental I: Influencia en los algoritmos de la existencia de retardos y errores en el funcionamiento y medidas de los sensores 127
 - Grupo Experimental II: Influencia de la duración del retardo en el consumo de memoria y número de operaciones 129

C4.1.2. Análisis de los resultados..... 130

- Análisis comparativo de los valores estimados por los diferentes algoritmos (grupo experimental I) 130
 - Comparación de los resultados obtenidos para el mismo ejemplo por los tres algoritmos..... 131
 - Influencia del desorden de las medidas en los resultados obtenidos 134
 - Importancia de la existencia de la etapa de validación cuando hay medidas erróneas. 140
 - Evolución del valor del estado y varianza para cada instante de muestreo según se va asimilando la información medida en instantes anteriores y que llega al algoritmo temporalmente retardada 142
- Análisis del consumo de memoria y de las necesidades de computo (grupos experimentales I y II) 145
 - Grupo Experimental I: Influencia en los algoritmos de la existencia de retardos y errores en el funcionamiento y medidas de los sensores 146
 - Grupo Experimental II: Influencia del retardo en el consumo de memoria y número de operaciones..... 148

C4.2. ANÁLISIS DEL FUNCIONAMIENTO DE LOS ALGORITMOS PARA SISTEMAS CON MODELOS

NO LINEALES 151

C4.2.1. Análisis del consumo de memoria y de las necesidades de cómputo..... 152

C4.2.2. Análisis de los resultados obtenidos por el algoritmo C3-3 (IFAsynMixto) sobre distintos sistemas no lineales 157

- Análisis de los resultados para un sistema con funciones no lineales suaves 157
 - Análisis de la influencia del desorden de las medidas en los resultados obtenidos y en su evolución según se asimilan las medidas desordenadas 160
 - Necesidad del término corrector de la medida para la asimilación sensorial proveniente de los sensores no lineales..... 164
 - Necesidad del recálculo de la información sensorial asociada a los sensores no lineales cuando llega al algoritmo información retardada proveniente de cualquier sensor..... 166

▪ Análisis de los resultados para un sistema con funciones no lineales abruptas	170
Necesidad del recálculo de la información sensorial asociada a los sensores no lineales	
cuando llega al algoritmo información retardada proveniente de cualquier sensor	173

CAPITULO V

UTILIZACIÓN DEL SISTEMA DE FUSIÓN EN UNA SISTEMA DE CONTROL EN TIEMPO REAL PARA ROBOTS AUTÓNOMOS MÓVILES..... 177

C5.1. INCORPORACIÓN DEL ALGORITMO IFASYNMIXTO A UNA ARQUITECTURA ROBÓTICA DE CONTROL	179
C5.1.1. Arquitecturas de control existentes.....	181
▪ Arquitecturas deliberativas o jerárquicas	181
▪ Arquitecturas reactivas.....	182
▪ Arquitecturas Híbridas	183
C5.1.2. Metodología de implementación.....	185
C5.1.3. Descripción de la arquitectura robótica.....	188
▪ ModeloRobot: Módulo principal de la arquitectura de control del robot.....	190
▪ ProgControl: Módulo de control, planificación, navegación y supervisión.....	192
Teclado: Módulo interfaz entre el usuario y el programa de control	194
Planificador: Módulo de planificación de caminos y vigilancia del entorno	194
Navegante: Módulo de seguimiento y supervisión de trayectorias.....	195
Control: Módulo de generación de las señales de control de los motores del robot	197
▪ ElRobot: Módulo utilizado para interactuar con un robot o simular su comportamiento	198
▪ Fusion: Módulo de integración y fusión multisensorial.....	202
Entorno: Módulo de fusión multisensorial de las medidas de entorno	203
Posición: Módulo de fusión multisensorial para estimar la posición del robot.....	207
C5.2. RESULTADOS OBTENIDOS POR EL ALGORITMO IFASYNMIXTO EN LA ARQUITECTURA ROBÓTICA PROPUESTA.....	209
C5.2.1. Resultados simulados.....	210
▪ Experimento simulado sin sensores	213
▪ Experimento simulado con medidas de los Encoders	215
▪ Experimento simulado con medidas de los Encoders y de la Brújula.....	217
▪ Experimento simulado con medidas de todos los sensores sin obstáculo en el mapa de objetos utilizado en la simulación	220
▪ Experimento simulado con medidas de todos los sensores con obstáculo en el mapa de objetos utilizado por el módulo ElRobot.....	227

Índice

C5.2.2. Resultados experimentales sobre el robot real	232
Experimento real en un pasillo: desplazamiento del robot a lo largo del mismo	234
Experimento real en el hall: desplazamiento del robot y esquivas de obstáculos	238
CONCLUSIONES	247
RESULTADOS Y CONCLUSIONES	249
a. Desarrollo de algoritmos de fusión para la asimilación secuencial y asíncrona de medidas retardadas y temporalmente desordenadas	249
b. Diseño e implementación de un sistema unificado de integración y fusión multisensorial de medidas temporalmente desordenadas para robots autónomos móviles.....	252
LÍNEAS DE TRABAJO FUTURO.....	253
PUBLICACIONES REALIZADAS	254
BIBLIOGRAFÍA	257
APENDICE I	
EXPRESIONES DEL FILTRO DE KALMAN Y DEL FILTRO DE INFORMACIÓN	277
A1.1. LOS CUATRO TIPOS DE SISTEMAS DE CONTROL PROPUESTOS	279
A1.2. EL FILTRO DE KALMAN Y DEL FILTRO DE INFORMACIÓN.....	280
A1.2.1. Predicción	281
A1.2.2. Asimilación.....	282
▪ Sistemas lineales con un único sensor.....	282
▪ Sistemas lineales multisensoriales	283
▪ Sistemas no lineales con un único sensor.....	284
▪ Sistemas no lineales multisensoriales	285
A1.2.3. Validación	285
A1.2.4. Tablas con las ecuaciones de los filtros de Kalman y de Información asociados a cada tipo de sistema.	285
APENDICE II	
ALGORITMOS PROPUESTOS PARA ESTIMAR EL ESTADO DE SISTEMAS DE CONTROL CON SENSORES ASÍNCRONOS Y MEDIDAS RETARDADAS Y DESORDENADAS	289
A2.1. TABLAS RESUMEN CON LOS ALGORITMOS PROPUESTOS	291

APENDICE III

EL ROBOT MÓVIL AUTÓNOMO NANDANA: DESCRIPCIÓN HARDWARE Y SOFTWARE	297
A3.1. DESCRIPCIÓN DEL ROBOT MÓVIL AUTÓNOMO NANDANA	299
A3.2. ARQUITECTURA DE CONTROL	302
A3.2.1. Arquitectura Hardware Distribuida de Bajo Nivel basada en el PIC16F873/6.....	304
A3.2.2. Comunicación entre el PC y los Microcontroladores: Protocolo Serie con Auto Identificación del Receptor	305
A3.2.3. Comunicación entre el PC Fijo y el PC Portátil: Sockets TCP/IP.....	308
A3.2.4. Comunicación entre los microcontroladores y Dispositivos I2C: Protocolo de Comunicación Serie I2C	308
A3.3. SISTEMA MOTRIZ.....	308
A3.4. SENSORIZACIÓN	311
A3.4.1. Encoders Ópticos Incrementales.....	312
A3.4.2. Brújula Electrónica.....	313
A3.4.3. Cinturón de Infrarrojos.....	315
A3.4.4. Cinturón de Ultrasonidos.....	317
A3.4.5. Sistema de Visión	322
A3.4.6. Sistema Orientable Triular de Ultrasonidos.....	324
A3.5. SOFTWARE DE CONTROL ASOCIADO A LOS PIC16F873/6: REPERTORIO DE INSTRUCCIONES	325

APENDICE IV

LA METODOLOGÍA ROOM Y LA HERRAMIENTA DE DISEÑO EDROOM.....	329
A4.1. LA METODOLOGÍA ROOM	331
A4.1.1. Actor.....	331
A4.1.2. Protocolos de Comunicación y Puertos.....	333
A4.1.3. Envío de Mensajes.....	333
A4.1.4. Servicios adicionales de ROOM	334
A4.1.5. Notación gráfica	334
A4.2. LA HERRAMIENTA DE DISEÑO EDROOM.....	337
A4.2.1. Editor Gráfico.....	338
A4.2.2. Generador de Código.....	339

A4.2.3. Librería EdROOM para Windows 2000	340
▪ Pr_task.....	342
▪ Pr_Semaphore	342
▪ Pr_Time.....	342
▪ Pr_Kernel	343
▪ Lista_tarea	343
A4.3. CONSTRUCCIÓN DE ARQUITECTURA SOFTWARE ROBÓTICA CON ROOM Y EdROOM	343
A4.3.1. Tipo de Arquitecturas Software Robóticas y ROOM	344
A4.3.2. Versatilidad de la Arquitecturas Robóticas diseñadas con ROOM y EdROOM	346

APENDICE V

MODELO DEL SISTEMA Y DE LOS SENSORES UTILIZADOS PARA ESTIMAR LA POSICIÓN DEL ROBOT AUTÓNOMO MÓVIL.....	347
A5.1. MODELO DE TRANSICIÓN DE ESTADOS: ECUACIONES DE MOVIMIENTO DEL ROBOT	349
A5.2. MODELOS DE MEDIDA: ECUACIONES DE LOS SENSORES UTILIZADOS PARA DETERMINAR LA POSICIÓN DEL ROBOT	351
A5.2.1. Brújula Electrónica.....	351
A5.2.2. Encoders ópticos incrementales.....	355
A5.2.3. Cinturón de ultrasonidos y mapa del entorno.....	355
A5.3. VERSIÓN ADAPTABLE DEL ALGORITMO IFASYNMIXTO	359

Introducción

La aparición de la fusión multisensorial como un concepto y disciplina con carácter propio, a finales de los años 80, ha favorecido la transferencia de terminología, conocimientos y algoritmos entre los campos de conocimiento donde se utiliza la información procedente de distintos sensores para llevar a cabo determinadas tareas. Este continuo intercambio de tecnología ha permitido el desarrollo de nuevos algoritmos de fusión, la búsqueda de nuevas áreas de aplicación de las técnicas desarrolladas, y la aparición de nuevos problemas a resolver.

Uno de estos problemas surge cuando se recibe información asíncrona, retardada, y/o temporalmente desordenada en el sistema de fusión. El origen del problema se encuentra en los subsistemas de intercambio de información de los sistemas de fusión distribuidos (a nivel sensorial, algorítmico, etc.), y en el uso de sistemas sensoriales asíncronos y algoritmos de procesamiento previos que retrasan la disponibilidad de la información existente.

Cuando esto ocurre en sistemas de fusión secuenciales que combinan cíclicamente la información sensorial asociada a sistemas dinámicos (cuyo estado puede ser modificado con el paso del tiempo) es necesario, por una parte, utilizar en los algoritmos de fusión elementos que permitan detectar la existencia del problema, y conveniente, por otra, incorporar mecanismos que permitan utilizar este tipo de información asíncrona, retardada y desordenada de forma automática.

La detección del problema es una labor sencilla, que se puede resolver añadiendo a cada dato el instante de tiempo al que éste se encuentra referido y usar estos tiempos para realizar comparaciones. Sin embargo, la utilización de medidas asíncronas, retardadas y desordenadas en los sistemas de fusión secuenciales requiere incorporar mecanismos más complejos, y es el origen del reciente desarrollo de diferentes algoritmos de asimilación asíncrona de medidas retardadas y temporalmente desordenadas, entre los que se encuentran los tres algoritmos que se proponen en esta tesis.

Nuestro interés por los algoritmos de asimilación asíncrona procede de un estudio, realizado al inicio de la tesis, sobre posibles mejoras del sistema de integración y fusión multisensorial para robots autónomos móviles desarrollado en el Departamento de Arquitectura de Computadores y Automática de la UCM ([Lopez-Orozco1999]). Ese sistema estaba compuesto por dos subsistemas de fusión independientes, uno encargado de estimar la posición del robot y otro la situación del entorno cambiante que le rodea, que además se encontraban implementados sobre dos técnicas de fusión diferentes. El subsistema de localización del robot se fundamentaba en un filtro de Kalman distribuido y asíncrono, capaz de trabajar con sensores con distintas velocidades de muestreo y de manejar pequeños retardos de comunicación entre los distintos nodos del filtro. El subsistema relacionado con la situación del entorno utilizaba un bosque de redes bayesianas discretas junto a un algoritmo de inferencia exacta, para construir un mapa probabilístico de celdas de ocupación del entorno en el que se mueve el robot. Además, el sistema de fusión estaba implementado de forma modular como parte de una arquitectura de control en tiempo real, cuyos módulos se comunicaban a través de un sistema de intercambio de mensajes basado en prioridades, motivo por el cual

algunas de las medidas llegaban retardadas y desordenadas a los algoritmos de fusión y no podían ser usadas correctamente.

La utilización de dos técnicas diferentes dificultaba la interconexión de los dos sistemas de fusión, motivo por el cual en el estudio se decidió desarrollar un sistema nuevo de fusión cuyos dos subsistemas se encontrasen, al menos, unificados bajo el uso de una técnica común. Un examen en profundidad del modelado de sistemas mediante redes bayesianas y de los algoritmos de inferencia probabilística existentes para las mismas, hace que las redes bayesianas sean la técnica elegida para unificar, en esta tesis, el modelado del sistema de fusión del robot.

Además, en el estudio realizado sobre las redes bayesianas se observó, por una parte, la equivalencia existente entre algunos algoritmos de inferencia probabilística para redes bayesianas y algoritmos de fusión secuencial clásicos como el filtro de Kalman (y el filtro de Información), y por otra, la capacidad de los algoritmos de inferencia para introducir la evidencia en cualquier orden. La problemática observada en la arquitectura de fusión y control existente, y la unión de las dos observaciones anteriores, es el origen de nuestra búsqueda de soluciones al problema general de la fusión secuencial y asíncrona de medidas retardadas y temporalmente desordenadas.

En la búsqueda de soluciones a este problema se tuvo en todo momento presente que los métodos propuestos debían ser versátiles, generales, y utilizables dentro de sistemas de control con limitaciones temporales significativas. Las primeras dos restricciones, versatilidad y generalidad, permitirán utilizar estos métodos para estimar el estado de diferentes sistemas dinámicos, mientras que la tercera hizo que durante el desarrollo de los algoritmos se optimizasen sus necesidades de cómputo y de memoria.

La versatilidad de los algoritmos desarrollados se pondrá de manifiesto en un análisis exhaustivo de su funcionamiento sobre diferentes sistemas simulados. Su idoneidad para estimar la posición de un robot se ha comprobado durante su uso en una nueva versión de la arquitectura de control de robots autónomos móviles, que ha sido desarrollada como parte del trabajo de esta tesis. El uso de los algoritmos propuestos en el subsistema de localización del robot ha mejorado al subsistema de localización original, ya que permite tratar de forma automática los retardos, desordenes y asincronismos de las medidas de este subsistema.

Además, con el objeto de que todo el sistema de fusión, y no sólo el subsistema de localización, fuese capaz de dar un tratamiento automático a los asincronismos, retardos y desordenes de las medidas, se (1) ha modificado (y optimizado) el algoritmo de inferencia utilizado sobre la red bayesiana del sistema que estima el estado del entorno, y (2) se ha establecido en el sistema de fusión un flujo de información consistente con la existencia de retardos en los dos subsistemas de fusión.

Los experimentos finales de los nuevos algoritmos y arquitectura de control se realizan sobre una plataforma robótica autónoma móvil, que se ha diseñado y construido a lo largo de la tesis para disponer de un banco de pruebas de algoritmos de fusión multisensorial.

Objetivos

A la hora de construir un sistema de fusión es conveniente diseñar o utilizar métodos de fusión multisensorial generales y versátiles, y usar técnicas de implementación que favorezcan la estructuración y comprensión del funcionamiento del sistema. Esta filosofía de trabajo permitirá adaptar el sistema de fusión desarrollado a futuras modificaciones del sistema sensorial, incrementar la funcionalidad del mismo a través de la inclusión de nuevas capas, y/o reutilizar los algoritmos desarrollados para resolver nuevos problemas. Por lo tanto, la búsqueda de generalidad y versatilidad permanecerá subyacente durante el desarrollo de los algoritmos y sistemas de fusión de esta tesis.

Los dos objetivos principales de la tesis son:

- El desarrollo de métodos generales de fusión secuencial para la asimilación asíncrona de medidas retardadas y temporalmente desordenadas.
- El diseño e implementación de un sistema unificado de integración y fusión multisensorial con medidas temporalmente desordenadas para el control de robots autónomos móviles.

Ambos objetivos se encuentran fuertemente entrelazados, ya que el desarrollo de métodos generales de fusión secuencial permitirá disponer de algoritmos de este tipo para el diseño e implementación del sistema de fusión de la arquitectura de control, y el diseño e implementación del sistema de fusión permitirá probar los algoritmos desarrollados cuando forman parte de un sistema de control complejo y real.

A continuación se detallan las metas particulares perseguidas y asociadas a cada uno de estos dos objetivos.

a. *Desarrollo de métodos de fusión secuencial para la asimilación asíncrona de medidas retardadas y temporalmente desordenadas*

El reciente desarrollo de diferentes algoritmos de fusión secuencial para la asimilación asíncrona de medidas retardadas y temporalmente desordenadas tiene como objeto solucionar las carencias de las soluciones clásicas al problema, consistentes en: (1) desechar la información retardada; (2) retardar el proceso de fusión; o (3) almacenar las medidas y cada vez que se recibe una nueva medida, reiniciar el proceso secuencial desde el instante de muestreo asociado a la misma.

Los tres algoritmos propuestos, que han sido desarrollados de forma simultánea a otras soluciones recogidas en la literatura con las que serán comparados tras su presentación, se caracterizarán por permitir:

- *El modelado del comportamiento del sistema mediante una red bayesiana*, debido a la potencialidad inherente a esta técnica de fusión y a la capacidad de algunos de sus algoritmos para inferir el estado de las variables a partir de la evidencia introducida en cualquier orden. Esta elección permitirá, además, unificar bajo el uso de una sola metodología al sistema de fusión de la arquitectura de control, cuando se implementen los algoritmos desarrollados en su subsistema de localización.
- *La obtención en todo momento de la “mejor” estima del estado del sistema (aquella que utiliza toda la información sensorial disponible, sin eliminar la retardada ni retardar el proceso de fusión), a pesar de que la información pueda llegar al sistema de forma asíncrona, retardada y desordenada*, y de que no exista información a priori sobre el asincronismo de los sensores ni sobre los retardos y desordenes de las medidas recibidas.
- *Su incorporación en sistemas de control en tiempo real*, objetivo para el que habrá que minimizar, en la medida de lo posible, el número de operaciones que realizan y sus necesidades de memoria.
- *El uso de técnicas de validación de las medidas* en una etapa previa a la asimilación de las mismas, *para poder asegurar un funcionamiento robusto de los algoritmos ante fallos de los sistemas de medida*.
- *Una fácil reconfiguración de los algoritmos desarrollados, para favorecer el uso de diferentes tipo y número de sensores, y la reutilización de los algoritmos para estimar el estado de distintos sistemas dinámicos de control*. Este objetivo conlleva la necesidad de incluir en la formulación de los algoritmos la existencia de señales de control y de múltiples sensores, así como de minimizar el número de suposiciones que restrinjan el rango de aplicabilidad de los algoritmos.

Para poner de manifiesto la generalidad y robustez de los algoritmos desarrollados se analizará su funcionamiento sobre diferentes sistemas simulados y sobre un sistema de fusión y control en tiempo real para robots autónomos móviles.

b. Diseño e implementación de un sistema unificado de integración y fusión multisensorial con medidas temporalmente desordenadas para robots autónomos móviles

Este objetivo supondrá una mejora del sistema de integración y fusión multisensorial para robots autónomos móviles desarrollado en el Departamento de Arquitectura de Computadores y

Automática de la Universidad Complutense de Madrid, que se encuentra descrito en [Lopez-Orozco1999].

Los objetivos de diseño e implementación de la nueva arquitectura de fusión y control son los siguientes:

- *Diseñar una arquitectura unificada de fusión, en la que los dos subsistemas de fusión, el de localización y el del entorno, se encuentren implementados utilizando una técnica común: las redes bayesianas.* La metodología elegida facilitará, en trabajos posteriores, el incremento de la funcionalidad del sistema de fusión.
- *Permitir la asimilación secuencial de medidas asíncronas, retardadas y temporalmente desordenadas dentro de todo el sistema de fusión.* Para el subsistema de localización del robot se utilizarán los algoritmos secuenciales desarrollados a lo largo de la tesis, cuyo funcionamiento será analizado, mediante simulaciones, antes de ser incorporados dentro de la arquitectura de fusión y control. El subsistema de fusión de medidas de entorno utilizará una nueva red y un nuevo algoritmo de inferencia para permitir también la asimilación de información asíncrona, retardada y temporalmente desordenada.
- *Conseguir una arquitectura versátil y portable, en la que se puedan probar fácilmente diferentes estrategias de control y fusión, y que pueda ser utilizada sobre diferentes plataformas robóticas.* Para conseguir estos objetivos se utilizará una herramienta gráfica que reduzca la separación existente entre las etapas de diseño y especificación, y la etapa de implementación final como un programa de control de tiempo real. Además, se rediseñarán las relaciones existentes entre los diferentes módulos del sistema y la funcionalidad de alguno de ellos.
- *Por último, se comprobará el funcionamiento de la nueva arquitectura y algoritmos de fusión sobre una plataforma robótica móvil* que ha sido diseñada y construida como parte del trabajo de esta tesis para disponer de un banco de pruebas de algoritmos de fusión multisensorial.

Organización de la memoria del trabajo realizado

La principal aportación de esta tesis es el desarrollo de algoritmos generales de fusión multisensorial secuenciales para sistemas con medidas asíncronas, retardadas y desordenadas. Por lo tanto, a lo largo de la exposición del trabajo realizado se dará un peso mayor a los algoritmos desarrollados que al diseño e implementación del sistema de fusión multisensorial y control de robots autónomos móviles. Sin embargo, este orden preferencial de la exposición de los algoritmos frente a la presentación de la nueva arquitectura de fusión no resta valor al trabajo realizado en este segundo campo.

La memoria ha sido dividida en 5 capítulos y 5 apéndices, cuyo contenido se detalla en las próximas secciones.

a. Resumen del contenido de los capítulos

En el capítulo I se introduce al lector en la problemática y los aspectos más relevantes de la fusión multisensorial, y en el funcionamiento de las redes bayesianas como técnica de fusión. Durante el estudio de la fusión multisensorial, como un marco de trabajo formal, se ponen de manifiesto los diferentes factores que determinan las técnicas y algoritmos más convenientes para resolver problemas concretos; se introduce el problema de la fusión de medidas asíncronas, retardadas y temporalmente desordenadas; y se presentan brevemente algunos de los objetivos del uso de la fusión en robots autónomos móviles. En el análisis de las redes bayesianas se recogen las características más relevantes del modelado de sistemas mediante esta técnica, se estudian los algoritmos existentes para inferir el estado de las variables del sistema en función de la evidencia, y se presentan las ventajas del uso de esta metodología como técnica de fusión multisensorial. A lo largo del capítulo se recoge el estado del arte, con las contribuciones más importantes y recientes relacionadas con los distintos apartados del capítulo.

En el capítulo II se presenta la red bayesiana elegida para modelar el comportamiento de sistemas dinámicos sobre los que se realizan medidas de forma asíncrona que pueden llegar retardadas y temporalmente desordenadas, y se desarrolla un algoritmo capaz de estimar el estado del sistema. Para realizar esta última labor se elige y modifica un algoritmo de inferencia exacto, minimizando su número de operaciones, comprobando que es capaz de asimilar medidas desordenadas, y analizando el porqué es capaz de asimilarlas. Además, se demuestra la equivalencia existente entre el algoritmo desarrollado y el filtro de Información (y por lo tanto el de Kalman), cuando las medidas recibidas y asimiladas en cualquier orden por el algoritmo desarrollado son proporcionadas de forma ordenada y sin retardos al filtro de Información. También se estudia la incorporación de una etapa de validación en el modelado del sistema y se proponen diferentes algoritmos que permiten incorporar esta etapa. El algoritmo presentado al final del capítulo permite estimar el estado de sistemas lineales dinámicos multisensoriales con variables gaussianas, incluye una etapa opcional de validación previa a la asimilación de las medidas, y obtiene, a pesar de los retardos y desordenes existentes en el proceso de recepción, los mismo resultados que el filtro de Información (y el de Kalman) cuando a este último se le proporcionan los mismos datos ordenados y sin retardos.

En el capítulo III se modela el comportamiento del sistema con una red bayesiana dinámica y se desarrollan los tres algoritmos propuestos en esta tesis para resolver el problema de la fusión secuencial de medidas asíncronas, retardadas y temporalmente desordenadas. El primero, llamado Bayes, se obtiene tras sustituir la red bayesiana del capítulo anterior por una red bayesiana

dinámica y adaptar el algoritmo presentado al final del capítulo anterior a la nueva red. El segundo, IFAsyn, es desarrollado con el objeto de disminuir las necesidades de memoria y cómputo del algoritmo Bayes y se fundamenta en los estudios del capítulo anterior sobre el proceso de asimilación de medidas ordenadas y desordenadas, y sobre la equivalencia existente entre los algoritmos desarrollados y el filtro de Información. IFAsyn es una nueva versión del filtro de Información que permite la asimilación de medidas asíncronas, retardadas y desordenadas. En el capítulo se muestra la equivalencia existente entre el algoritmo de *Bayes e IFAsyn (que sirven para estimar el estado sistemas dinámicos lineales con variables gaussianas aunque las medidas sean recibidas en el sistema de fusión retardadas y temporalmente desordenadas)*, y las mejoras de memoria y de cómputo que supone el algoritmo IFAsyn frente a Bayes. El tercer algoritmo, IFAsynMixto, surge ante la necesidad de ampliar el rango de aplicabilidad de los algoritmos a sistemas no lineales (o mixtos) y se realiza como una generalización simultánea del algoritmo IFAsyn y del filtro de Información Extendido. *IFAsynMixto* permite *estimar el estado de sistemas dinámicos multisensoriales con variables gaussianas, incluye una etapa opcional de validación previa a la asimilación de las medidas, y obtiene, a pesar de los retardos y desordenes existentes en el proceso de recepción, los mismo resultados que el filtro de Información Extendido (y el de Kalman Extendido) cuando a este último se le proporcionan los mismos datos ordenados y sin retardos*. Por último, al final del capítulo, se comparan los tres algoritmos desarrollados con otros algoritmos propuestos recientemente en la literatura para resolver problemas similares.

En el capítulo IV se analiza exhaustivamente el comportamiento de los algoritmos propuestos en el capítulo III, mediante simulación, sobre distintos tipos de sistemas. Durante los análisis realizados se ponen de manifiesto distintas características de los algoritmos desarrollados, y se realiza un examen del consumo de memoria y las necesidades de cómputo de los mismos.

En el capítulo V, que se encuentra dividido en dos partes, se presenta la arquitectura de fusión y control desarrollada, y los resultados experimentales de los algoritmos de fusión propuestos cuando forman parte de dicha arquitectura. En la primera parte del capítulo se presenta la metodología seguida para implementar la nueva versión de la arquitectura, y se realiza una descripción de la misma, en la que se pone de manifiesto su versatilidad y diseño orientado a la prueba de diferentes algoritmos de fusión y control sobre diferentes sistemas robóticos. Durante la descripción de los tres módulos de fusión de la arquitectura, que permiten la fusión asíncrona, retardada y desordenada de las medidas del sistema, se introduce el nuevo subsistema de fusión del entorno. En la segunda parte del capítulo se recogen los resultados obtenidos por los algoritmos de fusión del sistema, analizando en mayor profundidad los obtenidos por el subsistema de localización del robot, cuando la arquitectura de control es utilizada sobre un robot simulado y sobre el robot real.

Por último, se presenta un capítulo de conclusiones en el que se pone de manifiesto las principales aportaciones de las tesis, y las principales propiedades de los algoritmos y sistema de fusión desarrollados.

b. *Resumen del contenido de los apéndices*

En el apéndice I se comparan y recogen, en un conjunto de tablas, las ecuaciones de diferentes versiones del filtro de Kalman y del filtro de Información.

En el apéndice II se presentan de forma conjunta los algoritmos para la fusión secuencial de medidas asíncronas, retardadas y desordenadas presentados a lo largo del capítulo III, y cuyo funcionamiento es analizado sobre diferentes sistemas en los capítulos IV y V. Además, se presenta una versión aproximada del algoritmo IFAsynMixto, que reduce las necesidades de cómputo y memoria de la versión original, y que puede ser utilizada sobre algunos sistemas no lineales.

En el apéndice III se describe la plataforma robótica móvil diseñada y construida a lo largo de la tesis, utilizada como banco de pruebas para los algoritmos de fusión multisensorial desarrollados durante la misma, y para realizar otros trabajos de investigación.

En el apéndice IV se describen la metodología de diseño e implementación ROOM, y la herramienta EdROOM, que son utilizadas para definir la arquitectura de control y fusión.

Y en el apéndice V se presentan las ecuaciones que modelan el comportamiento del robot y de los sensores utilizados dentro del subsistema de fusión de localización de la arquitectura de control y fusión. Además, se recoge una versión adaptada del algoritmo IFAsynMixto que tiene en cuenta las peculiaridades del robot y sus sensores, y que es usada en las pruebas finales sobre el robot real.

Capítulo I

Fusión Multisensorial y Redes Bayesianas

En este capítulo se muestran los aspectos más relevantes de la fusión multisensorial y se ponen de manifiesto algunos de los problemas asociados a los métodos y herramientas pertenecientes a este marco formal. A lo largo del mismo se presentan las ventajas del uso combinado de diferentes fuentes de información para resolver problemas concretos, y se relacionan los factores que determinan las técnicas y algoritmos más convenientes para cada problema. Entre estos factores se pueden destacar el tipo de información que se fusiona, y el tratamiento y el uso que se hace de la información. Otro aspecto a tener en cuenta es el tipo de arquitectura utilizado a la hora de implementar los diferentes procesos del sistema de fusión.

El elevado y creciente número de técnicas existentes hacen que, aun teniendo en cuenta todos estos factores, la elección de una técnica de fusión para resolver problemas concretos no sea una labor sencilla y que los mismos problemas estén siendo resueltos por diferentes autores utilizando diferentes técnicas. Para evitar la necesidad de rediseñar un nuevo algoritmo cuando se introducen cambios en alguna parte del sistema (por ejemplo, en el subsistema sensorial) será conveniente diseñar y utilizar técnicas generales, que permitan de forma sencilla modificar o ampliar la funcionalidad del sistema, o reutilizarlo para otros problemas similares.

Una de estas técnicas generales, que permiten realizar un diseño incremental de sistemas de fusión multisensorial la constituyen las redes bayesianas. Su uso como técnica de fusión está creciendo en los últimos años debido a sus características, que serán analizadas detalladamente en este capítulo. Entre las mismas, cabe destacar el permitir un modelado gráfico de las relaciones de dependencia de las variables del sistema, el utilizar una definición compacta de sus relaciones probabilísticas, y el permitir diferentes tipos de razonamiento sobre las variables de la red. Por estas y otras razones, expuestas en el capítulo, es la técnica elegida para implementar un sistema unificado de fusión multisensorial para robots autónomos móviles.

La problemática en la que se centran los algoritmos recogidos en el sistema desarrollado, también se introduce, brevemente, en dos apartados de este capítulo. Por una parte se encuentran los objetivos del sistema de fusión de los robots autónomos móviles, que son conocer en todo momento la posición del robot y determinar el estado de su entorno. Por otra, el tipo de información que se va a fusionar, medidas asíncronas, retardadas y temporalmente desordenadas, y la forma de tratar este tipo de información dentro de los algoritmos de fusión secuenciales.

El capítulo se encuentra organizado en dos secciones. En la primera se presentan los aspectos y problemática general de la fusión e integración multisensorial; se introduce el problema de la fusión de medidas asíncronas, retardadas, y temporalmente desordenadas; y se presentan los objetivos principales de los sistemas de fusión multisensorial en robots autónomos móviles. En la segunda se realiza una descripción general de las redes bayesianas, en la que se ponen de manifiesto aquellas propiedades que hacen que además de ser una técnica clasificable como método de fusión, esté siendo utilizada cada vez en un mayor número de sistemas.

C1.1. Integración y Fusión Multisensorial

La integración multisensorial se centra en el estudio de la utilización conjunta de la información proporcionada por los diferentes dispositivos sensoriales de un sistema, con el objeto de realizar una tarea determinada. El uso de múltiples fuentes de información es ventajoso porque [Luo1989]:

- La información *redundante*, o características observadas simultáneamente por diferentes sensores, permite mejorar la fiabilidad del sistema y reducir su incertidumbre total.
- La información *complementaria*, o distintas propiedades observadas por diferentes sensores, permite construir modelos más completos sobre la realidad del sistema.
- El *incremento de la cantidad de información disponible* en todo instante aumenta la oportunidad de razonar a partir de dicha información, y por lo tanto, la velocidad de respuesta del sistema.
- Habitualmente el *coste económico* de un sistema multisensorial es *menor* que el de un sistema equivalente formado por un único sensor que sea capaz de proporcionar las mismas prestaciones.

La fusión multisensorial, por otra parte, es una parte opcional de los sistemas de integración multisensorial, cuyo objeto va más allá de la utilización conjunta de la información proporcionada por diferentes sensores. *Los procesos de fusión son aquellos en los que la información, proporcionada por diferentes fuentes en un instante determinado (fusión multisensorial) y/o por el mismo sensor a lo largo del tiempo (fusión temporal), es combinada bajo un formato de representación único.*

El origen de la información proporcionada a los procesos de fusión e integración no tiene que ser necesariamente sensorial, ya que éstos permiten combinar información proveniente de otro tipo de fuentes, como por ejemplo el conocimiento a priori que se tiene del sistema observado.

Estas definiciones establecen claramente las diferencias existentes entre los procesos de integración y fusión multisensorial: (1) el proceso de fusión queda enmarcado como un tipo especial de proceso de integración y/o como una etapa dentro del mismo, y (2) el proceso de integración puede ser llevado a cabo sin realizar ningún tipo de fusión, hecho que ocurre en un amplio conjunto de sistemas multisensoriales en los que la información sensorial es sólo utilizada de forma conjunta para realizar una determinada tarea y/o la información de unos sensores sirve para guiar el funcionamiento de otros.

Los procesos de fusión multisensorial confieren a los sistemas de integración multisensorial en los que se incluyen de dos características fundamentales: (1) reducen la dimensionalidad del problema

al combinar la información procedente de las distintas fuentes de información en un único formato de representación y (2) aumentan la capacidad de abstracción del sistema ya que bajo dicho formato de representación se puede incluir una descripción del sistema observado más exacta, más completa, y menos ligada a las fuentes de información.

C1.1.1. Fusión Multisensorial

La fusión de datos es un marco formal en el que se expresan los métodos y herramientas que permiten la unión de datos originados por diferentes fuentes con el objeto de extraer de dicha unión información de “mayor calidad” (concepto dependiente de la aplicación para la que la información será utilizada). En el caso en el que las fuentes de datos sean sensores, el marco formal se denomina fusión multisensorial. Esta definición general, recogida en [Wald1998] y [Wald1999], pone énfasis en el marco formal y en los fundamentos de la fusión, y no en los métodos, algoritmos, operadores, niveles, arquitecturas, etc., como sucede en definiciones más clásicas ([Pau1988], [JDL1991], [Hall1992], [Luo1992], [Houzelle1994], [Mangolini1994], [Hall1997]). Aún así, estas definiciones ponen de manifiesto diferentes aspectos de la fusión, por lo que serán analizadas a lo largo de esta sección.

En [Hall1992] la fusión multisensorial se define como un proceso de elevada complejidad de realización que depende de un elevado número de factores. Las técnicas y algoritmos más convenientes en cada caso vienen determinados por:

- El *tipo de aplicación* para la que se van a utilizar los resultados del proceso de fusión, que determina: (1) el fenómeno que hay que observar, (2) los sensores e información a priori utilizados como fuentes de información, y (3) el tipo de razonamiento que se realizará sobre la información recibida por el sistema de fusión.
- La *incertidumbre* asociada a la información y la posible *discrepancia* entre la información proveniente de distintas fuentes, que tienen que ser tratados de acuerdo con las características intrínsecas de dicha información y el tipo de razonamiento que se quiera establecer.
- La *velocidad de respuesta* mínima necesaria para el buen funcionamiento del sistema, factor de crucial importancia en los sistemas de *control en tiempo real*. Ésta se encuentra directamente relacionada con la velocidad del subsistema de integración y, por lo tanto, con el tiempo máximo de respuesta de los procesos de fusión y la velocidad de muestreo de las fuentes sensoriales.
- Las *características propias de la arquitectura de control del sistema y la localización y características de los sensores*, así como el mecanismo de *comunicación e intercambio de información entre los distintos subsistemas*, hecho especialmente importante en los casos en los que se encuentren distribuidos.

Todos estos factores son el origen de un amplio conjunto de arquitecturas y algoritmos de fusión multisensorial. La fuerte dependencia existente entre el algoritmo desarrollado y las características anteriormente expuestas, especialmente las relacionadas con los sensores utilizados, puede reducir la capacidad de generalización de los sistemas de fusión e impedir la reutilización de los algoritmos desarrollados para una determinada aplicación en otras.

Uno de los objetivos principales de esta tesis es el desarrollo de algoritmos reutilizables para otros sistemas y aplicaciones (similares y/o diferentes), por lo que durante su definición se evitará, en la medida de lo posible, tomar decisiones que aumenten la dependencia entre nuestro sistema, un robot autónomo móvil, y los algoritmos de fusión utilizados.

Como primera fase del trabajo de investigación analizaremos los principales tipos de procesos de fusión existentes, presentando en la siguiente sección las clasificaciones habituales de los mismos: en función del tipo de información que combinan, de acuerdo con el tratamiento y uso que hacen de la misma, y según el tipo de algoritmo y arquitectura de control utilizada. Estas clasificaciones no deben ser entendidas como particiones rigurosas, sino como una forma de exponer los diferentes objetivos de los algoritmos, los tipos de información que fusionan, la organización del sistema, etc. Tras el análisis se presenta el problema sobre el que se centra el desarrollo de los algoritmos propuestos en esta tesis: el asincronismo, retardo y desorden de las medidas en los sistemas de fusión. Por último, se establece la relación existente entre la fusión multisensorial y otras áreas de conocimiento, se pone de manifiesto el creciente interés existente por la fusión, y se presentan algunos de los problemas asociados a los robots móviles autónomos que están siendo solucionados utilizando técnicas de fusión.

▪ **Clasificación de los procesos de fusión**

La gran variedad de procesos de fusión multisensorial tiene su origen en las diferencias existentes entre las aplicaciones para los que se usan, los tipos de datos medidos, el tratamiento que se le da a los mismos, la arquitectura de control del sistema en el que se incorporan, etc. Cada uno de estos factores es utilizado como una característica discriminante en las diferentes clasificaciones de los algoritmos de fusión presentados en las siguientes secciones.

Nivel funcional: tratamiento y uso que se hace de la información

La necesidad de dividir los procesos de fusión desde el punto de vista del tratamiento y uso que se hace de la información surgió a finales de los años ochenta, momento en el que la fusión de datos comienza a ser entendida como una disciplina con carácter propio, con el objeto de unificar la terminología existente hasta ese momento y proporcionar un marco de trabajo común [White1991]. La clasificación resultante, inicialmente presentada en [White1988] y actualizada con un nivel adicional en [Hall1997], constituye un modelo genérico de proceso de fusión formado por un

conjunto de 4 niveles. Esta clasificación, esquematizada en la figura C1-1, no debe de ser entendida como una partición rigurosa en la que un proceso únicamente puede realizar labores de un nivel, sino como una forma de distinguir los diferentes objetivos del sistema de fusión.

En el esquema de la figura C1-1, para cada nivel se identifica el tipo de información sobre la que razona, el tipo de procesamiento realizado, y el tipo de información de entrada y salida utilizada por los algoritmos pertenecientes a dicho nivel. Los niveles se encuentran ordenados de acuerdo con la complejidad del tipo de razonamiento que realizan. De abajo a arriba, el nivel I se ocupa de identificar y localizar los objetos y fenómenos de interés; el nivel II es el responsable de realizar una descripción de la situación y las relaciones existentes entre los objetos y sucesos del entorno, el nivel III realiza un análisis conjunto de la situación y predice las intenciones del sistema y los objetos observados, y por último, el nivel IV supervisa y optimiza el funcionamiento del resto de los niveles. Los cuatro niveles interaccionan mediante un intercambio continuo de información, no necesariamente jerárquico (de un nivel a su superior). Además de los cuatro niveles, se ha representado su interacción con los otros módulos de la arquitectura de control de sistema: (1) el módulo sensorial que proporciona las medidas y (2) el sistema de control de alto nivel que toma decisiones a partir de las inferencias realizadas por el sistema de fusión multisensorial.

Los algoritmos desarrollados en esta tesis, que son utilizados dentro del sistema de fusión de una arquitectura de control de un robot para su localización, así como el algoritmo modificado a lo largo de la misma para estimar el estado del entorno pertenecen al primer nivel de esta clasificación, aunque utilizan técnicas clasificables en el segundo.

Nivel de representación: tipo de información que combinan

La información recibida por el proceso de fusión puede ser utilizada como el factor discriminante utilizado para la clasificación de los sistemas de fusión. De acuerdo con dicho factor, en [Luo1992] los procesos de fusión pueden ser clasificados en cuatro niveles: de señal, de píxel, de características y simbólico. El tipo de información almacenada en cada nivel, y el grado de abstracción asociada a la misma, determina el tipo de procesamiento realizado y los sistemas que pueden incorporar ese tipo de fusión. En la figura C1-2 se presentan algunas de las características asociadas a cada uno de los cuatro niveles de fusión.

Al igual que en la sección anterior, en los sistemas de fusión reales no existe una división clara entre los diferentes niveles. Además, la información proporcionada por un sensor o subsistema de fusión puede ser utilizada por subsistemas de fusión pertenecientes a diferentes niveles.

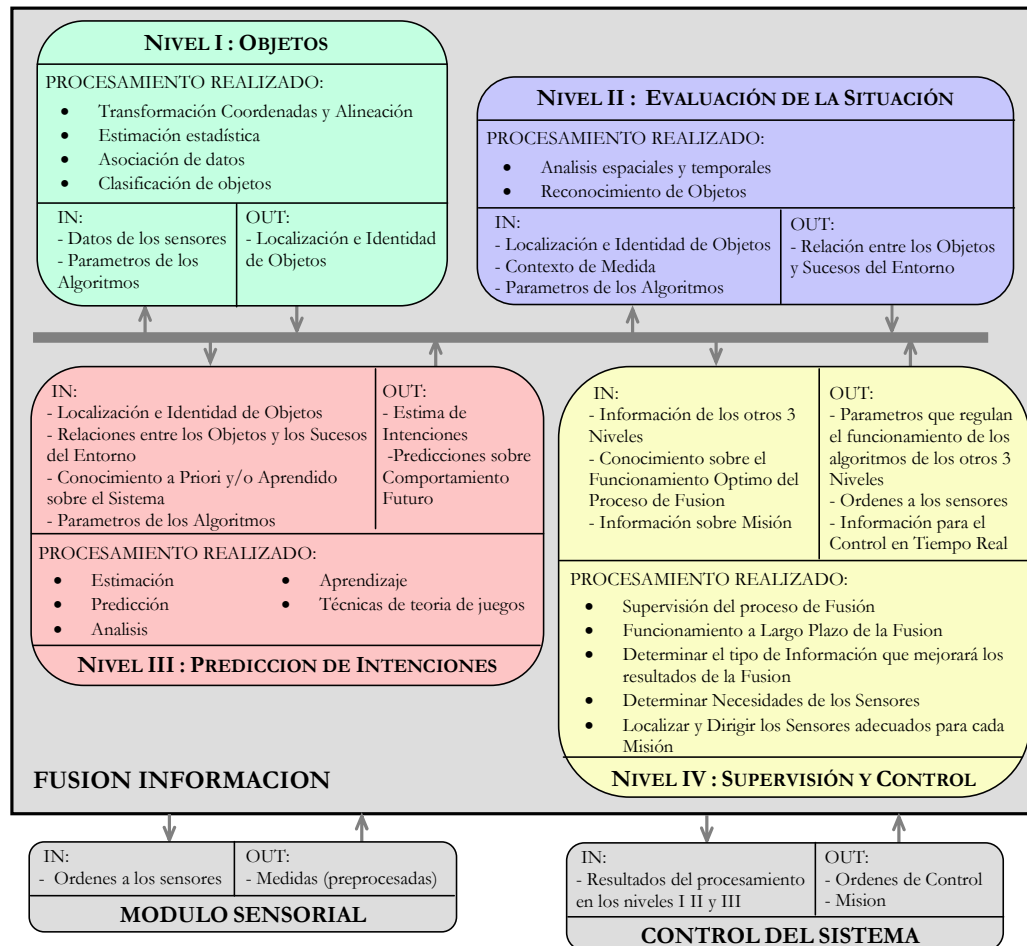


Figura C1-1. Clasificación Funcional de los Procesos de Fusión (definición JDL).

NIVEL SEÑAL	<p>¿QUÉ ? : Señales eléctricas sincronizables proporcionadas por los sensores (datos en bruto). La información puede provenir de varios sensores (fusión multisensorial) y/o del mismo sensor en distintos instantes de tiempo (fusión temporal)</p> <p>¿CÓMO?: Utilizando <i>técnicas de procesado de señal</i> y realizando suposiciones sobre las señales (ruido de medidas, dependencia existentes entre las mismas)</p> <p>¿DÓNDE?: Técnicas especialmente utilizables en <i>sistemas de tiempo real</i> y <i>sistemas empostrados</i></p>
NIVEL PIXEL	<p>¿QUÉ ? : La información almacenada en <i>píxeles</i> (unidad mínima de información visual). Se puede fusionar la información relativa a un pixel proporcionada por distintas señales para formar una imagen (fusión a nivel de señal), la información relativa a un pixel almacenada en varias imágenes (fusión multisensorial y/o temporal) y/o la información de un conjunto de píxeles de la misma imagen (fusión espacial).</p> <p>¿CÓMO?: Utilizando <i>técnicas de procesamiento de señal, procesado de imágenes, segmentación, extracción de características</i>, ... Es conveniente utilizar técnicas que permitan el procesamiento paralelo y/o que optimicen el procesamiento local a nivel de pixel</p> <p>¿DÓNDE?: <i>Sistemas sin restricciones temporales con una elevada capacidad de procesamiento</i></p>
NIVEL SIMBÓLICO	<p>¿QUÉ ? : Información abstracta de alto nivel obtenida por sensores dispares o relativa a regiones del entorno diferentes.</p> <p>¿CÓMO?: Utilizando <i>métodos de razonamiento simbólico</i> que hacen uso de información a priori almacenada en un modelo propio del sistema y que manejan la incertidumbre asociada a los diferentes elementos abstractos.</p> <p>¿DÓNDE?: <i>Sistemas sin restricciones temporales</i>, como consecuencia del tiempo previo asociado a la extracción de la información abstracta y al procesamiento de este nivel</p>
NIVEL CARACTERÍSTICA	<p>¿QUÉ ? : Características del sistema (información vectorial extraída de los datos sensoriales por algún tipo de procesamiento previo), como son propiedades de los objetos, estado del sistema, con el objeto de disminuir la incertidumbre asociada a los mismos</p> <p>¿CÓMO?: Utilizando <i>técnicas de estimación, asociación, análisis espacial y temporal</i>,...</p> <p>¿DÓNDE?: <i>Sistemas con restricciones temporales intermedias</i>, ya que aunque el ancho de banda asociado a la comunicación de características es pequeño, el tiempo previo utilizado en su extracción puede no serlo.</p>

Figura C1-2. Clasificación de los Procesos de Fusión a Nivel de Representación

Los algoritmos desarrollados en esta tesis, que son utilizados dentro del sistema de fusión de una arquitectura de control de un robot para localizar el robot, así como el algoritmo modificado a lo largo de la misma para estimar el estado del entorno se pueden englobar en el nivel de características de esta clasificación.

Nivel algorítmico: técnicas utilizadas para fusionar la información

Una clasificación íntimamente relacionada con las dos clasificaciones anteriores (ligada al procesamiento realizado en el esquema del nivel funcional de la figura C1-1 y a cómo se realiza la fusión en el esquema del nivel de representación de la figura C1-2) es aquella que divide a los procesos de fusión según el tipo de teoría y el algoritmo sobre el que se soportan. La mayoría de los algoritmos existentes se pueden clasificar en una de las dos teorías siguientes: teoría de la detección o teoría de la estimación:

- La teoría de la detección (o decisión) compara la información (por ejemplo, las medidas proporcionadas por los sensores) de la que se dispone sobre un sistema con un conjunto de hipótesis para decidir cual de ellas describe mejor su situación. Dentro de esta teoría se engloban un amplio conjunto de métodos probabilísticos, como son la inferencia clásica ([Peebles1987], [Papoulis1990]), la inferencia bayesiana ([Duda1979]), los factores de certeza ([Shortliffe1975], [Buckanan1984], [Heckerman1986]), las redes bayesianas ([Pearl1988], [Shafer1990a], [Heckerman1995], [JensenFV1996], [Castillo1997]), la teoría de la evidencia de Dempster-Shafer ([Dempster1967], [Shafer1976], [MurphyRR1998]), la teoría generalizada de Thomoupoulos ([Thomopoulos1990]); y métodos heurísticos como la lógica borrosa ([Zadeh1975], [Zadeh1978]), las redes neuronales ([Bishop1995], [Haykin1999]) o las máquinas de vectores soporte ([Vapnik1998], [Cherkassky1998], [Vapnik2000]). Algunos métodos se pueden entender como un caso particular de otros y en algunos algoritmos se combinan varios métodos ([Lauritzen1996b]).
- La teoría de la estimación es una extensión del proceso de decisión mediante el cual se calcula el valor de un parámetro utilizando los valores de múltiples medidas, relacionadas de alguna forma con dicho parámetro. Este cálculo se realiza buscando el mejor ajuste del parámetro a los datos observados, suponiendo para este fin que se conoce inicialmente las fuentes de información disponibles, la existencia de modelos que relacionan las medidas con el parámetro (y en algunos casos, que establecen el comportamiento del parámetro a lo largo del tiempo), que existe un conocimiento estadístico de los errores de las medidas (y del comportamiento del parámetro), etc. Los métodos existentes se diferencian en el criterio de optimización elegido (mínimos cuadrados, mínimos cuadrados ponderados, máxima verosimilitud, etc.), en el método de optimización elegido para implementar el criterio (gradiente, quasi-Newton, algoritmos genéticos, Newton-Raphson, etc.), y en el método de

procesamiento (por lotes o secuencial). Ejemplos clásicos de este tipo de algoritmos, que se pueden encontrar en [Gelb1974], [Anderson1979], [Bar-Shalom1993] y [Bar-Shalom2001], son: mínimos cuadrados de Gauss (1795), máxima verosimilitud de Fisher (1912), la estimación estadística de Kolmogorov y Weiner (años cuarenta), el estimador de Kalman ([Kalman1960], [Kalman1961]) o sus equivalentes en el espacio de información ([Mutambara1998]), los filtros alfa-beta ([Kalata1984]), etc.

Al igual que en las clasificaciones anteriores no se debe considerar que existe una división estricta entre ambas teorías, ya que existen algoritmos clasificables dentro de las dos. Por ejemplo, las redes bayesianas, que en su versión discreta pueden ser utilizadas para decidir cuales de las hipótesis que describen un sistema es la que mejor se ajusta a los datos observados ([Lopez-Orozco1998a]), o que en su versión continua pueden ser utilizadas para estimar las posiciones de objetos ([MurphyKP1998b], [Thrun2002]).

Además de los algoritmos clasificables en las dos metodologías expuestas, existe un grupo de algoritmos que realizan labores auxiliares, no por eso menos importantes en los procesos de fusión. Entre los mismos se encuentran algoritmos de alineación de datos, preprocesamiento, manipulación de bases de datos, asociación paramétrica, etc.

Los algoritmos desarrollados y modificados en esta tesis pueden englobarse dentro de las dos teorías.

Nivel de implementación: arquitectura de control del sistema de fusión

Y por último, pero no por eso menos importante, la decisión de dónde se combina la información procedente de cada una de las fuentes determina, durante el proceso de implementación, la estructuración del proceso de fusión. Las arquitecturas de control centralizadas, en las que se fusionan directamente los datos en bruto provenientes de los sensores, y las descentralizadas, en las que se realiza un procesamiento local dividido en diferentes módulos, constituyen las dos posibilidades extremas de organización, que son aplicables a los diferentes niveles funcionales ([Waltz1990]) y de representación. A medio camino, y con el objeto de aprovechar las ventajas de cada una de las posibilidades extremas, se sitúan las arquitecturas híbridas. Las dos arquitecturas extremas se caracterizan por:

- La arquitectura centralizada, al utilizar directamente los datos brutos proporcionados por los sensores, puede extraer la máxima información de los mismos, aunque para ello sobrecargue el sistema con fuertes necesidades de cálculo y de comunicación (de los datos en bruto), y complejos modelos físicos.
- La arquitectura descentralizada, también denominada autónoma, divide la funcionalidad en procesamiento locales correspondientes a diferentes módulos (o etapas), de los que se

abstraen características más generales utilizables en otros módulos. Esta división reduce el flujo de datos, a costa de ir perdiendo información según la abstracción de los resultados aumenta y de incrementar la dependencia entre los resultados obtenidos por cada módulo.

La implementación de algoritmos de fusión generales y reutilizables para distintos sistemas lleva de forma natural a la implementación de arquitecturas híbridas, que aprovechan las ventajas de abstracción de las arquitecturas autónomas y la riqueza de la información disponible en las arquitecturas centralizadas. La arquitectura utilizada para nuestro sistema robótico será, por lo tanto, de este último tipo.

▪ ***El problema de la recepción de medidas asíncronas, retardadas y temporalmente desordenadas en el sistema de fusión***

La implementación de sistemas distribuidos de fusión, la utilización de sensores con diferentes velocidades de muestreo, el uso de algoritmos de preprocesado, y la fusión de información procedente de otros subsistemas, hace que una parte de la información pueda ser recibida en el sistema de fusión de forma asíncrona, retardada y temporalmente desordenada.

Este tipo de recepción no suele ser un problema para los sistemas que estiman el estado del sistema con métodos de procesamiento por lotes, ya que la fusión no se realiza hasta que toda la información se encuentra disponible, y la información puede ser ordenada antes de empezar el proceso de fusión.

Sin embargo, los métodos de procesamiento secuenciales actualizan el estado del sistema tan pronto como el algoritmo de fusión recibe nueva información sobre el sistema, y el hecho de que ésta pueda encontrarse retardada y/o temporalmente desordenada, debe ser contemplado por el algoritmo, y resuelto de alguna forma. La falta de mecanismos automáticos para tratar con información retardada y temporalmente desordenada en los métodos clásicos, hace que las primeras soluciones aportadas para el tratamiento de este tipo de información se basen en:

- Eliminar la información retardada, y por lo tanto cualquier tipo de desorden. Esta solución es aceptable cuando solo unas pocas medidas sufren el problema del retardo, aunque en cualquier caso, se estará eliminando información que el sistema de fusión debería haber utilizado para disminuir la incertidumbre que se tiene sobre el estado del sistema. El crecimiento de la incertidumbre que se tiene sobre el sistema, ante la pérdida de información es analizada, para el filtro de Kalman, en [Sinopoli2004].
- Esperar un tiempo, antes de realizar la fusión, para que la información retardada pueda llegar antes de fusionar la información asociada a instantes posteriores que ya ha sido recibida. Esta solución requiere que el sistema sepa de la existencia de medidas, así como del valor de sus retardos, y pospone el cálculo de la “mejor estima” (aquella que usa toda la información

disponible) hasta que se reciba la información retardada. El sistema de fusión para robots autónomos móviles del que se partió en esta tesis, utiliza esta solución en el subsistema de localización del robot ([Lopez-Orozco2000]).

- Almacenar las variables estimadas y toda la información sensorial para poder reiniciar el proceso de fusión desde el instante en el que la información retardada debió ser recibida. Esta solución permite obtener la “mejor estima” en todo momento, a costa de sobrecargar computacionalmente el sistema de fusión con operaciones que habrá que realizar de nuevo, pero en las que no se había podido incluir, al no estar disponible, la información retardada.

Durante los últimos años se han buscado otras soluciones al problema, orientadas a evitar la eliminación de información y los tiempos de espera, y reducir el número de operaciones y/o las necesidades de memoria. Ejemplos de las mismas, para estimar el estado de un sistema dinámico en el último instante de muestreo a partir de toda la información disponible hasta dicho instante pueden encontrarse en [Orton2001], [Nettleton2001], [Mallick2001], [ZhangK2002], [Bar-Shalom2002a], [Bar-Shalom2002b], [Challa2003], [ZhangK2003] y [Mookerjee2004].

En esta tesis se proponen tres algoritmos para solucionar el mismo problema, dos para sistemas lineales y uno para no lineales, desarrollados de forma paralela a los anteriormente mencionados. Al final del capítulo III se analizan las semejanzas y diferencias entre los algoritmos desarrollados en esta tesis y los existentes en la literatura.

▪ **Aplicaciones de fusión**

La fusión multisensorial no es una técnica reciente. Es utilizada de forma natural por los seres vivos, que a lo largo de su evolución, han desarrollado la capacidad de utilizar combinadamente la información procedente de diferentes sensores para sobrevivir. Además, desde hace décadas se ha estado realizando en diferentes campos de conocimiento, aunque sin identificar los procesos realizados como técnicas de fusión multisensorial.

Sin embargo, la aparición de la fusión como un concepto y disciplina con carácter propio se remonta a finales de los años 80. Ésta se fundamenta en la existencia de nuevos sensores y potente hardware de procesamiento, y en el interés de las agencias de Defensa de EEUU por crear un marco de trabajo común en el que se estandarice la terminología y los principios de diseño, y se analicen las técnicas existentes en diferentes campos de conocimiento (procesamiento de señales, estimación estadística, control de sistemas, inteligencia artificial, métodos numéricos), etc.

La terminología y algoritmos desarrollados inicialmente como aplicaciones militares, se han extendido a otras áreas de conocimiento, como son la medicina, robótica, aeroespacial, meteorología, geofísica, etc. El tratamiento de la fusión como una disciplina en si misma ha

permitido que la transferencia de conocimientos y tecnología se realice también en sentido inverso (de las aplicaciones no militares a las militares).

El número de sistemas identificados como aplicaciones que incorporan métodos de fusión crece continuamente, así como el interés que existe por los mismos. Esto se pone de manifiesto en la existencia de numerosos libros ([Durrant-Whyte1988], [Hager1990], [Waltz1990], [Hall1992], [Abidi1992], [Luo1994], [Manyika1994], [Varshney1997], [Brooks1998], [Bouchon-Meunier1999], [Hall2001], [Zhu2002], [Klein2004], [Hall2004], etc.), publicaciones y congresos especializados en la materia (Information Fusion [InformationFusion]; the 7th Internacional Conference on Information Fusion [Fusion2004]; the Internacional Conference on Intelligent Sensors, Sensor Networks and Information Processing [ISSNIP'04]; conferencias del SPIE en Sensor Fusion [SPIE_SensorFusion], Remote Sensing [SPIE_RemoteSensing] y Defense&Security [SPIE_Defense&Security], etc.) y artículos en publicaciones de los diferentes campos de conocimiento que utilizan técnicas de fusión multisensorial.

Fusión en robots autónomos móviles

Un sistema robótico autónomo se define como un agente “inteligente” que percibe lo que sucede en su entorno mediante el uso de un conjunto de sensores y responde o actúa sobre el mismo por medio de un conjunto de actuadores [Russell1995]. Su grado de autonomía real se encuentra íntimamente ligado a la complejidad de los sistemas sensorial y de actuadores, así como de las relaciones existentes entre ambos sistemas. Estas relaciones vienen determinadas por la capacidad del robot para extraer información de los datos medidos, aprender de la misma, hacer un uso simultáneo de dicha información y del conocimiento inicial incorporado al sistema, y actuar de acuerdo con las conclusiones extraídas. *En el caso de los robots autónomos móviles dicha autonomía esta directamente relacionada con su capacidad para determinar la posición en la que se encuentra y el estado del entorno que le rodea.*

Por lo tanto, *dentro del área de los robots autónomos móviles*, sistema sobre el que se probarán los algoritmos de fusión secuencial de medidas asíncronas, retardadas y temporalmente desordenados desarrollados a lo largo de este trabajo de investigación, *los objetivos principales del sistema de fusión son estimar la posición del robot y el estado del entorno.*

El problema de la localización del robot, fundamental para lograr un cierto grado de autonomía ([Cox1991]), *se puede presentar de diferentes formas* ([Fox2001]). El problema más sencillo, conocido como *seguimiento de posición*, es el de determinar la posición del robot en todo instante conocida la posición inicial y las medidas proporcionadas por los sensores. Un problema más complejo es el de *la localización global*, en el que hay que determinar la posición del robot sin conocer su posición inicial. Y otro problema diferente, todavía más difícil de resolver, es el del *robot raptado*, en el que el sistema de localización del robot considera que el robot se encuentra en

una posición diferente a la real. Para resolver el primer problema se pueden utilizar técnicas clásicas de seguimiento de objetos ([Bar-Shalom2001]), mientras que para resolver con éxito los otros dos se están utilizando técnicas mas recientes, como los filtros de Partículas ([Thrun2002]).

El conocimiento del estado del entorno es fundamental para asegurar la reactividad del robot ante los cambios que se producen en el mismo. La construcción de un mapa del mismo, es una solución al problema, en la que se puede representar la presencia y el tipo de objetos, las relaciones existentes entre los mismos, sus intenciones, etc. Una simplificación del problema consiste en dividir el espacio en un enrejado de celdas a las que se le asignan diferentes características, y fusionar la información que se tenga sobre las mismas ([Elfes1990], [Borenstein1991], [Singhal1997], [Lopez-Orozco1998a]).

La solución simultánea de la localización del robot y construcción de un mapa del entorno es un problema adicional, en el que se establece un flujo de información e incertidumbre entre los dos problemas básicos ([Thrun1997], [Castellanos2001], [Montemerlo2002]). Este problema se puede resolver con dos subsistemas de fusión diferentes, que intercambian información, o utilizando un único sistema, aunque en este caso es habitual resolver el problema en dos etapas sucesivas (en las que se supone conocida la posición del robot o de los objetos del entorno). En cualquier caso, el uso de una técnica de fusión única en los dos subsistemas o etapas del algoritmo favorecerá el flujo de información entre las dos etapas o subsistemas, y un posterior incremento de su funcionalidad.

El sistema de fusión inicialmente disponible en el Departamento de Arquitectura de Computadores de la UCM, descrito en [Lopez-Orozco1999], resuelve el problema por medio de dos subsistemas de fusión diferentes en los que el flujo de información únicamente se realiza en un sentido, del sistema de localización al sistema de construcción al mapa de entorno. Cada uno de los mismos, utiliza una técnica de fusión diferente, por lo que se ha decidido, como primer objetivo de esta tesis, unificar el sistema de fusión existente por medio utilizando la misma técnica en los dos subsistemas. Esta unificación del sistema se realizará utilizando diferentes redes bayesianas, capaces de modelar gráficamente las relaciones existentes entre las variables involucradas en el proceso de fusión, y cuyos algoritmos permiten realizar diferentes tipos de razonamiento y cálculos probabilístico sobre el sistema.

Obviaremos realizar una enumeración detallada de los numerosos sistemas de fusión multisensorial para robots autónomos móviles que resuelven cada uno de los problemas anteriormente expuestos, para centrarnos directamente en la descripción de la técnica probabilística elegida y justificar las causas de nuestra elección.

C1.2. Redes Bayesianas

Entre las diferentes técnicas probabilísticas que se pueden utilizar para implementar sistemas de fusión multisensorial se encuentran las redes bayesianas (también denominadas redes bayesianas de creencia [Pearl1986], redes probabilísticas causales [Andreassen1987], redes causales [Lauritzen1988], etc.). *Esta herramienta*, fundamentada sobre las observaciones de modularidad del teorema de Bayes realizadas por Pearl [Pearl1982], *permite representar de forma estructurada la probabilidad conjunta de un conjunto de variables aleatorias, utilizando para este fin un modelo gráfico de las relaciones existentes entre dichas variables*. Su origen bayesiano permite realizar cálculos sobre probabilidades subjetivas, tratar con la incertidumbre asociada a sistemas reales, incorporar en el razonamiento realizado el conocimiento a priori que se tiene sobre el sistema, ... siendo necesario cuantificar a priori parte de la información del sistema para llevar a cabo todas estas labores ([Papoulis1990], [Hearst1997]). *Aunque las redes bayesianas se utilizan habitualmente para inferir probabilísticamente los valores de algunas de las variables que contienen, también es posible realizar otro tipo de operaciones sobre las mismas: encontrar la configuración de variables más probable ([Cowell1992], [Nilsson1998], [JensenCS1998], [Campos1999]), medir posibles conflictos entre los valores de las variables de la red [Kim1995], realizar análisis de sensibilidad sobre la influencia de unas variables en otras ([JensenFV1995], [Gaag1999]), detectar que variables proporcionan información relevante para deducir la información de una variable determinada ([JensenFV1994a]), etc.* Además, se encuentra íntimamente relacionada con los diagramas de influencia ([Howard1984]) y las redes de decisión ([Cowell1994]), que permiten calcular estrategias de control óptimas por medio de la inclusión de otros tipos de nodos (de decisión y utilidad) a los propiamente bayesianos (que representan las variables aleatorias del sistema) en la estructura de la red bayesiana básica ([Pearl2000], [JensenFV2001b]).

El creciente interés por las redes bayesianas y la utilización de las mismas para la solución de problemas reales en diferentes campos de conocimiento se pone de manifiesto en la existencia de numerosos libros donde se presentan los fundamentos teóricos y/o uso práctico de las mismas ([Pearl1988], [Neapolitan1990], [Shafer1990a], [Gammerman1994], [Almond1995], [JensenFV1996], [Lauritzen1996a], [Shafer1996], [Gammerman1996], [Castillo1997], [Cowell1999], [Pearl2000], [JensenFV2001a], [Neapolitan2003], [Korb2003], etc.), las numerosas publicaciones especializadas existentes (proceedings of the AUA [UA12004], proceedings of the IPMU [IPMU2004], números especiales de la revista de Artificial Intelligence (www.aaai.org) y Comunicaciones del ACM (www.acm.org)), la gran variedad de artículos en diferentes campos de conocimiento, y el amplio conjunto de programas de distribución gratuita y comerciales (como los recogidos en www.ai.mit.edu/~murphyk/Bayes/bnsoft.html y [directory.google.com/Top/Computers/Artificial Intelligence/Belief Networks/Software/](http://directory.google.com/Top/Computers/Artificial_Intelligence/Belief_Networks/Software/)), etc.

En las secciones siguientes se analiza el funcionamiento de las redes bayesianas, cuando son utilizadas para inferir el estado de alguna de sus variables, y se justifica el uso de las mismas como técnicas de fusión.

C1.2.1. Funcionamiento de las Redes Bayesianas

Las redes bayesianas constituyen una técnica estructurada de razonamiento probabilístico, en el que es importante distinguir: (1) el modelado gráfico de las relaciones de dependencia entre las diferentes variables del sistema, (2) las relaciones probabilísticas que se establecen entre dichas variables, y (3) el conjunto de operaciones que se pueden realizar. Esos tres campos, que a su vez constituyen la base de conocimiento del problema (modelo gráfico + relaciones probabilísticas) y el motor de inferencia o razonamiento realizado, son presentados en las siguientes secciones. De las operaciones realizables, enumeradas en la sección anterior, únicamente trataremos las de inferencia probabilística (obtención de la distribución marginal posterior de las variables no observadas a partir de la evidencia).

En la siguiente figura se presenta una red bayesiana ejemplo, junto con las relaciones probabilísticas necesarias para inferir el estado de cualquier variable, y la operación sobre la que se fundamenta todo el cálculo probabilístico asociado a las redes bayesianas, para el caso de las variables y estructuras de la red ejemplo propuesta.

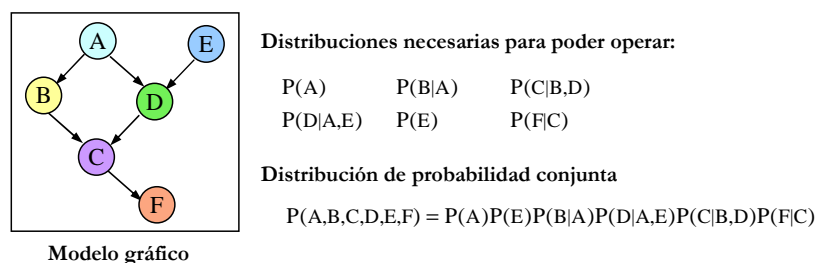


Figura C1-3. Ejemplo de red bayesiana, formada por 6 variables aleatorias (A, B, C, D, E y F). En la parte izquierda de la figura se presenta su módulo gráfico: un grafo dirigido y acíclico. En la parte derecha, se muestra las distribuciones necesarias para poder inferir el estado de cada una de sus variables y la expresión que representa la distribución de probabilidad conjunta para la red elegida.

▪ Modelado gráfico

Una de las características mas atrayentes de las redes bayesianas es la utilización de una descripción gráfica de la relaciones de dependencia de las variables que describen el sistema. Esta descripción se realiza sobre un grafo, cuyos nodos están asociados a las variables aleatorias y cuyos arcos capturan las relaciones existentes sobre las variables. En la parte izquierda de la figura C1-3 se representa el modelo gráfico de una red con 6 variables (A,B,C,D,E,F) relacionadas entre si según la estructura definida por los arcos del grafo. Los grafos correspondientes a las redes bayesianas cumplen dos restricciones principales sobre las que se fundamentan las operaciones

realizadas: tienen que *ser dirigidos y acíclicos*. Otros tipos de redes probabilísticas, como son las redes de Gibbs o las redes gráficas de cadena, admiten otro tipo de estructuras [JensenFV1999].

La utilización de grafos como herramienta de modelado, no sólo permite disponer de una descripción gráfica del sistema, sino que sirve como soporte para implementar los métodos de razonamiento aplicables a las redes bayesianas. Por ejemplo, el concepto de separación direccional permite determinar si la evidencia sobre un nodo de la red puede influir en el conocimiento que se tiene sobre otros nodos ([Pearl1988]).

Los enlaces existentes entre las variables no tienen porque representar impacto causal, ya que las propiedades asociadas al concepto de separación direccional, definido inicialmente sobre redes causales, forman parte de la semántica de la red [JensenFV1999]. Además, la existencia de un enlace entre dos variables tampoco implica que haya una dependencia condicional entre las mismas, ya que este tipo de dependencia obedece también a la distribución condicional existente entre las variables. Por lo tanto, la estructura de la red bayesiana y el concepto de separación direccional permiten determinar la falta de dependencia entre variables de la red, pero no la existencia de la misma [Neapolitan1990].

Existen diferentes tipos de redes bayesianas, clasificables, por ejemplo, según el tipo de variables asociadas a los nodos. *Las redes bayesianas discretas trabajan con variables discretas, las redes bayesianas continuas sobre variables continuas y las redes bayesianas mixtas con variables discretas y continuas. Las redes bayesianas dinámicas son aquellas cuya estructura evoluciona a lo largo del tiempo*, y sirven para modelar el comportamiento de sistemas donde el estado de una variable se ve modificado con el paso del tiempo. Casos particulares de sistemas cuyo comportamiento se puede modelar con una red bayesiana dinámica lo constituyen aquellos identificables como modelos ocultos de Markov (HMM), sistemas lineales dinámicos (LDS) o procesos de decisión de Markov parcialmente observables (POMDP).

Por último, destacar que en los casos en los que no se conoce la estructura de la red asociada a un sistema, ésta puede ser obtenida, a partir de las observaciones realizadas sobre el mismo, utilizando distintas técnicas de aprendizaje ([Buntine1994], [Heckerman1996], [Friedman1998], [Jordan1999]).

▪ **Relaciones probabilísticas**

La estructura de la red bayesiana, representada mediante un grafo dirigido y acíclico, no establece ninguna restricción sobre las relaciones probabilísticas de las variables que la conforman, ya que únicamente representa la parte cualitativa de la definición de la red. La parte cuantitativa la constituyen las definiciones probabilísticas de las relaciones existentes entre las variables. La restricción cuantitativa que hace que una distribución de probabilidades P sobre un conjunto de

variables $V = \{v_1, v_2, \dots, v_{n-1}, v_n\}$ pueda ser modelada mediante una red bayesiana, es que la distribución cumpla que cada variable v_i del grafo sea condicionalmente independiente de las variables asociadas a sus ancestros $an(v_i)$ cuando se conoce el valor de las variables asociadas a sus nodos padre $pa(v_i)$ (i.e. $P(v_i | W \cup pa(v_i)) = P(v_i | pa(v_i))$ con $W \subset an(v_i)$) [Neapolitan1990]. Por el ejemplo, para la red de la figura C1-3, se cumple que la variable C es condicionalmente independiente de A y E cuando se conoce el valor de las variables B y D.

La restricción de independencia condicional, junto con la propiedad de separación direccional, permite definir la probabilidad conjunta de todas las variables de la red función de la probabilidad condicional de cada variable y sus padres ($P(v_i | pa(v_i))$). En la ecuación (C1-1), conocida habitualmente como regla de la cadena, se recoge la expresión de probabilidad conjunta de las variables. En la parte izquierda de la figura C1-3 se representa la probabilidad conjunta de las variables de la red de la derecha.

$$P(V) = \prod_{v_i \in V} P(v_i | pa(v_i)) \quad (C1.1)$$

La posibilidad de definir la probabilidad conjunta a partir de las probabilidades condicionales $P(v_i | pa(v_i))$ es una característica sumamente ventajosa, ya que por una parte facilita la labor del usuario a la hora de definir las probabilidades necesarias para realizar los razonamientos posteriores, y por otra constituye una definición compacta de la que se puede obtener los resultados de los razonamientos.

Al igual que ocurre en el caso de la definición de la estructura, las probabilidades condicionales asociadas a las variables de la red pueden ser obtenidas, a partir de las observaciones del sistema, utilizando algoritmos de aprendizaje.

Por último, resaltar que en la definición del concepto de red bayesiana (como un conjunto $V = \{v_i \mid i=1:n\}$ de variables aleatorias, asociadas a los nodos de un grafo dirigido, sobre las que es necesario definir las probabilidades condicionadas $P(v_i | pa(v_i))$) no se establecen restricciones sobre el tipo de variables o de probabilidades se pueden utilizar. Las restricciones de este tipo son impuestas posteriormente, a la hora de desarrollar los diferentes algoritmos de razonamiento. Por lo tanto, entre las numerosas posibilidades existentes, habrá que elegir aquellos que permitan razonar sobre la red bayesiana utilizada para modelar el comportamiento de un determinado sistema.

▪ Algoritmos de inferencia probabilística

Los algoritmos que se presentan en esta sección son únicamente los relacionados con el problema de inferencia probabilística cuyo objetivo es estimar el estado de una variable (o conjunto de variables) en función del conocimiento a priori que se tiene sobre el sistema (las probabilidades condicionales) y la evidencia observada (conocimiento disponible sobre el estado de algunas

variables de la red). Dicho de otra forma, calcular las probabilidades marginales posteriores de las variables no observadas, a partir de la evidencia existente. Las operaciones de inferencia sobre redes bayesianas constituyen un problema NP complejo ([Cooper1990], [Dagum1993]), por lo que a la hora de diseñar la base de conocimiento (modelo gráfico + probabilidades condicionadas) del sistema es importante evitar, en la medida de lo posible, la existencia de múltiples conexiones ([Campos2000]).

Los algoritmos de inferencia existentes se pueden dividir en dos grandes grupos: los algoritmos de inferencia exactos y los algoritmos de inferencia aproximados. Los del primer grupo, permiten determinar de forma exacta las probabilidades marginales posteriores de las variables no observadas del sistema, mientras que los del segundo, utilizan métodos aproximados para determinar dichas probabilidades.

Una ventaja importante de los métodos exactos es que las probabilidades posteriores obtenidas no dependen del orden en el que se introduce la evidencia en la red. Aún más, se obtiene el mismo resultado final si se introduce toda la información y después se propaga, que si se introducen datos sucesivamente, y se realizan varias propagaciones. Este modo de proceder también puede ser realizado con algunos métodos aproximados, aunque el resultado obtenido por los mismos puede depender del orden en el que la evidencia ha sido introducida. Sobre esta característica se fundamenta una parte del desarrollo de los algoritmos propuestos a lo largo de esta tesis.

La ventaja principal de los métodos aproximados es su capacidad para resolver problemas sobre un conjunto de tipos de redes más amplio. En los métodos exactos desarrollados se imponen restricciones sobre las relaciones permitidas entre las diferentes variables (por ejemplo, que un nodo continuo no pueda tener descendientes discretos) y sobre el tipo de probabilidades condicionales (tablas de probabilidad discreta, gaussianas multivariantes, etc.), hecho que restringe el número de problemas que se pueden resolver con este tipo de algoritmos. Además, los métodos aproximados aceleran el proceso de inferencia para redes de elevada complejidad, incluso aquellas en las que es posible utilizar los algoritmos exactos.

Los dos tipos de algoritmos no constituyen una división infranqueable. En [Dawid1995] se propone un algoritmo para poder utilizar simultáneamente diferentes tipos de algoritmos de propagación, y alguno de los algoritmos existentes combina algoritmos básicos de los dos tipos ([Hernandez1997]).

Por último, es importante tener en cuenta que, además de los algoritmos especialmente desarrollados para redes bayesianas para realizar inferencias sobre el estado de sus variables, es posible en algunos casos utilizar métodos clásicos equivalentes. En el caso de optar por esta posibilidad, la base de conocimiento del sistema (modelo gráfico + probabilidades condicionales) será utilizada para modelar el comportamiento del sistema y el motor de inferencia utilizado será

un algoritmo no específico a la inferencia sobre redes bayesianas. *Parte de los algoritmos desarrollados en esta tesis y presentados en el capítulo III son englobables dentro de este subconjunto especial.*

En las secciones siguientes se relacionan algunos de los algoritmos existentes de cada una de las dos categorías básicas, al mismo tiempo que se analizan alguna de sus características.

Algoritmos de inferencia exacta

Los algoritmos de este tipo, utilizados para obtener la probabilidad posterior marginal de las variables de la red, se fundamentan en la existencia de expresiones exactas para las integrales de las densidades condicionales elegidas [Cowell1999]. Por este hecho, se restringe significativamente el número de redes y tipos de relaciones probabilísticas sobre los que se pueden utilizar. Los métodos que se presentan a continuación han sido inicialmente desarrollados para redes con variables discretas, y modificados en algunos casos para redes continuas y mixtas, como son las redes de variables continuas gaussianas relacionadas entre sí de forma lineal ([Lauritzen1992], [Lauritzen1999]).

El primero de los métodos propuestos ([Kim1983]) únicamente sirve para redes bayesianas con un modelo gráfico tipo poliárbol, y ha sido desarrollado sobre una arquitectura de intercambio de mensajes. A partir del mismo, y siguiendo dos filosofías diferentes se ha desarrollado un amplio conjunto de métodos para otros tipos de grafos dirigidos y acíclicos.

En la primera de las filosofías se construye una estructura secundaria (denominada árbol de unión) cuyos nodos están formados por conglomerados (subconjuntos) de las variables del modelo gráfico original, y se realizan un conjunto de operaciones sobre las funciones de probabilidad conjunta (denominadas potenciales) asociada a diferentes nodos de la nueva estructura. El tipo de operaciones realizadas así como el orden establecido en las mismas diferencian a los algoritmos desarrollados siguiendo este modo de proceder (Lauritzen-Spiegelhalter [Lauritzen1988], Hugin [JensenFV1990a], [JensenFV1990b], y Shafer-Shenoy [Shafer1990b]). El árbol de unión, construido moralizando y triangulando el grafo original, permite inferir el estado de cualquiera de sus variables y no preestablece ningún orden en la observación de la evidencia. Este hecho, impide a los algoritmos explotar las independencias introducidas por las variables observadas (extraíbles a partir de la definición de separación direccional) y por lo tanto, simplificar el cálculo en sistemas complejos en los que se conozca a priori los nodos observables y el instante de observación. Además, para redes bayesianas complejas el número de combinaciones de los estados de las variables de un conglomerado puede ser demasiado elevado para resolver el problema con los algoritmos de este tipo.

La segunda de las filosofías se fundamenta en el uso del concepto de separación direccional para simplificar el cálculo probabilístico del sistema. Una de estas técnicas, denominada condicionamiento, consiste en instanciar un grupo de las variables de la red, con el objeto de bloquear el paso de la información a través de las mismas. Si el número de variables instanciadas convierte el grafo en un poliárbol se pueden utilizar algoritmos de paso de mensajes similares al primer método propuesto ([Pearl1986], [Diez1996]). Otra de las técnicas consiste en determinar inicialmente el conjunto objetivo de variables sobre las que inferir (un subconjunto de todas las variables de la red), determinar el conjunto de variables de la red relevantes para llevar a cabo dichos razonamientos, y calcular la probabilidad final marginalizando las variables que no son objetivo del producto de las probabilidades condicionales de las variables relevantes ([Shachter1990b], [Dechter1996], [ZhangNL1996]).

Las dos filosofías anteriores pueden ser aplicadas simultáneamente para desarrollar algoritmos que aprovechen las ventajas inherentes a cada una. De esta forma surgen los algoritmos propuestos en [Madsen1998], [Madsen1999a] y [Madsen2000], que construyen un árbol de unión y mantienen en cada conglomerado una representación factorizada de las probabilidades. Antes de propagar la evidencia a través de la red, aplican el concepto de separación direccional dentro de cada conglomerado, para decidir que elementos de la representación factorizada es necesario transmitir.

Por último resaltar que el grado de paralelismo propio de las redes bayesianas y del cálculo probabilístico asociado a las mismas [Madsen1999b] permite implementar de forma paralela los algoritmos anteriormente expuestos y/o alguna de sus etapas. Ejemplos de algunos algoritmos que explotan el paralelismo existente son los propuestos en [Diez1994b], [Kovlov1994], [Kjærulff1997], [Kjærulff1998] y [Pennock1998]

Algoritmos de inferencia aproximada

Los algoritmos de este tipo basan su funcionamiento en la generación de un conjunto de N muestras (realizaciones) y en el cálculo de la probabilidad de ciertos sucesos (combinación de estados de conjuntos de variables de la red) a partir de los cocientes de frecuencia de aparición de los sucesos en la muestra generada y del tamaño de la población [Castillo1997].

Según el método utilizado para la generación de la muestra, se pueden clasificar como algoritmos estocásticos (generan la muestra de forma aleatoria) o deterministas (generan la muestra de forma sistemática). Ejemplos de algoritmos del segundo tipo son el muestreo sistemático ([Bouckaert1996]) y el método de búsqueda de la máxima probabilidad ([Henrion1991], [Poole1993], [Srivinikas1996]).

Los métodos estocásticos se diferencian unos de otros por: (1) las funciones de probabilidad utilizadas para generar las N realizaciones de la muestra, (2) las expresiones empleadas para

calcular el peso de cada elemento de la muestra en el cálculo de la probabilidad de los sucesos, y (3) la forma de introducir la evidencia en el cálculo de las muestras. A continuación se presentan algunos métodos en los que se pone de manifiesto dichas diferencias. Otros métodos pueden encontrarse en [Salmeron2000].

- En el método de muestreo lógico (también conocido como método de aceptación y rechazo o de muestreo hacia delante), [Henrion1988], se ordenan ancestralmente las variables de la red (padres antes que hijos), y siguiendo dicho orden se genera el valor de cada variable de una realización de acuerdo con la función de probabilidad $P(v_i | pa(v_i))$. Cuando hay evidencia, se eliminan las realizaciones contrarias a la misma. El peso de todas las realizaciones válidas es 1, y la probabilidad de un suceso se obtiene como la frecuencia de ocurrencia del mismo en las N realizaciones.
- En el método de muestreo de Markov (o de Gibbs) ([Pearl1987]) se le asignan los valores observados a las variables evidenciales y se genera aleatoriamente una muestra inicial. Seguidamente, se genera el valor de las variables no evidenciales, una a una y siguiendo un orden arbitrario (causal en el muestreo de Gibbs), de acuerdo con la función de probabilidad $P(v_i | pa(v_i)) \prod_{j \in \text{children}(v_i)} P(v_j | pa(v_j))$ y los estados del resto de las variables. Una vez que el proceso se realiza sobre todas las variables no evidenciales, se almacena la realización generada, y se comienza el proceso, utilizando como punto de partida, la última muestra obtenida. La probabilidad de un suceso se puede calcular como la frecuencia de dicho suceso en el conjunto de N muestras o como la medida de la probabilidad condicional del suceso en todas las realizaciones. En [JensenCS1995] se presenta una variante de este método, conocida como muestreo por bloques de Gibbs, en el que se muestrean conjuntos de variables.
- En el método de la función de verosimilitud pesante ([Fung1990], [Shachter1990a]) se le asignan los valores observados a las variables evidenciales y se realiza una ordenación ancestral de las variables no evidenciales. Siguiendo dicho orden, para cada realización se genera el valor de los estados de las variables no evidenciales con la función de probabilidad $P_e(v_i | pa(v_i))$. El peso de la realización en la muestra se calcula como $\prod_{e \in \text{Evidenciales}} P(v_e | pa(v_e))$.

Un conjunto adicional a los métodos estocásticos anteriormente expuestos lo constituyen los filtros de Partículas, cuyos algoritmos ([Gordon1993], [Kunazawa1995], [Kitagawa1996], [Isard1996], [Doucet2000], etc.) han sido diseñados para resolver el problema de inferencia en un tipo de redes bayesianas dinámicas: las cadenas de Markov. Los filtros de Partículas son métodos estocásticos secuenciales, que a partir del conocimiento de un conjunto de variables de la red y_t , estiman la distribución conjunta de otras variables x_t como un conjunto de N partículas $\{x_{t,i}^i | i=1:N\}$ (realizaciones sobre las variables estimadas) y N pesos $\{w_{t,i}^i | i=1:N\}$ (asociados a la probabilidad

relativa de cada partícula dentro del conjunto) para cada subred temporal t de la red dinámica. La secuencialidad del filtro se manifiesta en el método de cálculo de las N partículas y sus N pesos de cada subred temporal $\{x_t^i, w_t^i \mid i=1:N\}$ en función del conjunto de las subredes temporales anteriores $\{x_k^i, w_k^i \mid i=1:N, k=1:t-1\}$, las relaciones de independencia existentes en el modelo gráfico, las funciones de probabilidad condicionales de la red y la información existente de las variables evidenciales de la red y_t . La forma de resolver este problema no es única y es el origen de los diferentes filtros de Partículas existentes ([Doucet2001], [Ristic2004]).

Por último, es importante resaltar que durante la exposición de los métodos aproximados recogidos en este apartado, no se ha establecido ninguna restricción sobre el modelo de la red ni el tipo de relaciones probabilísticas de las variables. Por lo tanto, estas técnicas pueden ser utilizadas en un conjunto de redes más amplio que el de las técnicas de inferencia exacta. Aun más, los métodos aproximados aceleran el proceso de inferencia para redes de elevada complejidad, incluso aquellas en las que es posible utilizar los algoritmos exactos. El uso de técnicas de programación paralela permitirá acelerar los algoritmos aproximados en aquellas etapas en las que se puedan obtener realizaciones independientes.

C1.2.2. Redes Bayesianas y Fusión

Las redes bayesianas constituyen en si mismas técnicas de fusión de información, ya que permiten, a través de los cálculos de inferencia realizados sobre las mismas, combinar probabilísticamente (y teniendo en cuenta la incertidumbre asociada al sistema) la información que se tiene sobre distintas variables del sistema. En el caso en el que las fuentes de información sean sensoriales, la combinación realizada, puede ser directamente enmarcada dentro del ámbito de la fusión multisensorial.

Una de las propiedades mas atrayentes del uso de las redes bayesianas como técnicas de fusión multisensorial es la posibilidad de capturar gráficamente las relaciones de independencia existentes entre las variables del sistema. Además, durante la etapa inicial de modelado no es necesario determinar el tipo de funciones probabilísticas asociadas a las relaciones establecidas en la estructura, hecho que permitirá probar en una segunda etapa de diseño aquellas que se consideren apropiadas para modelar la incertidumbre y ruido inherente al sistema modelado, y elegir la más adecuada en cada caso. En la elección posterior de las probabilidades condicionales se puede incluir en la red el conocimiento a priori que se tiene sobre las relaciones probabilísticas de las variables y la incertidumbre asociada a dichas relaciones.

No existen restricciones iniciales sobre el tipo de variables asociadas a la red, hecho que permite realizar fusión en (y entre) los diferentes niveles de representación. Además, la estructura multinivel inherente a la red bayesiana de un sistema complejo ([Bangsø2000], [Xiang2000],

[Willsky2002]) favorece la construcción de una red a partir de múltiples subredes, por medio de la inclusión de enlaces que relacionen las variables de unas subredes con las de otras en los casos en los que se desee permitir el flujo de información entre las subredes. De esta forma, se puede conseguir un modelado incremental del sistema de fusión, en el que cada subred fusione localmente su propia información con la información procedente del resto del sistema.

Tras el modelado gráfico del sistema y la elección de las probabilidades condicionales asociadas a la red es necesario elegir el algoritmo más apropiado para llevar a cabo la fusión de información, entre los algoritmos disponibles para las relaciones estructurales y probabilísticas establecidas en la red bayesiana. Los algoritmos de inferencia exacta resultarán más apropiados en los casos en los que la red sea lo suficientemente sencilla y las relaciones estructurales y probabilísticas así lo permitan. Los algoritmos de inferencia aproximada serán aplicables en todos los problemas para los que no exista un algoritmo exacto o sea necesario reducir las necesidades de cómputo o de memoria asociada a los algoritmos exactos. Además, para algunas redes bayesianas particulares, siempre es posible utilizar algoritmos probabilísticos clásicos equivalentes a los algoritmos de inferencia desarrollados para cualquier tipo de red.

Asimismo, el uso de otro tipo de algoritmos *distintos* a los de inferencia sobre el estado de las variables de la red, permitirá que el algoritmo de fusión realice labores asociadas al nivel de supervisión y control de la clasificación funcional de los procesos de fusión. Por ejemplo, se podrá detectar conflictos entre los valores de las variables del sistema, determinar que sensores son capaces de proporcionar la información más relevante en cada caso o cuales permitan discriminar mejor diferentes estados de las variables, etc. Incluyendo nodos de decisión y utilidad en la red, se podrá utilizar ésta para además calcular estrategias de control.

Todas estas características hacen que las redes bayesianas sean la metodología elegida para unificar el sistema de fusión existente, que al inicio de esta tesis estaba compuesto por dos subsistemas implementados sobre dos técnicas diferentes.

▪ **Algunos ejemplos del uso de redes bayesianas como técnicas de fusión**

Todos los ejemplos de uso de redes bayesianas en los que se estima el estado de una o varias variables, pueden ser considerados un proceso de fusión, ya que el algoritmo de inferencia combina automáticamente el conocimiento del estado de las variables evidenciales de la red con la incertidumbre asociada al resto de las variables. En los ejemplos elegidos a continuación, se pone de manifiesto explícitamente esta relación y algunas características relevantes de las redes bayesianas en fusión multisensorial.

La diversidad de campos en los que las redes bayesianas son utilizadas y la variedad de problemas que son capaces de resolver con información generada por el mismo tipo de sensores demuestran los diferentes tipos de información que se pueden fusionar.

Dentro del área de visión es posible encontrar numerosos ejemplos de diferentes usos que a la información procedente del mismo tipo de sensor se le puede dar dentro del ámbito de las redes bayesianas. Por ejemplo, en [JensenFV1990c], [Jian-Ming1994] y [Dickison1997] se utilizan redes bayesianas multinivel junto con la información procedente de diversas regiones de una imagen para caracterizar el tipo de imagen (u objeto) que se está observando y dirigir el proceso de análisis de las regiones. En [Singhal2000] se propone una red bayesiana multinivel para detectar las zonas relevantes de una imagen. En [Regh1999], [Sherrah2000] y [Lan2002] las redes bayesianas son utilizadas para fusionar conocimiento contextual de alto nivel que se tiene sobre el sistema con diferentes atributos visuales de la imagen (colores, movimiento, texturas, etc.), en el primer caso para detectar cuando una persona está hablando, en el segundo para seguir objetos en movimiento, y en el tercero para monitorizar la fatiga humana. En [Dellaert1999] se localiza un robot a partir de medidas de la luminosidad del techo de las salas en las que se desplaza.

Otro campo de aplicación de las redes bayesianas lo constituye la robótica. Dentro del área de la robótica autónoma móvil podemos destacar los siguientes trabajos. En [Sahami1995] se presenta una arquitectura robótica cuyo módulo de control utiliza una red bayesiana para fusionar la información de un mapa cambiante para determinar las órdenes que se deben mandar a los actuadores. En [Singhal1997], [Berler1997], [Lopez-Orozco1998a] y [MurphyKP1999] se propone el uso de redes bayesianas para fusionar diferentes tipos de información sensorial sobre el entorno del robot para estimar (construir, “aprender”) un mapa del mismo. El modelado del problema de la localización mediante redes bayesianas se presenta en [Nicholson1994] (bajo la perspectiva de un problema de monitorización) y en [Fox2001] (bajo el prisma del uso de diferentes filtros de Partículas). Ejemplos de la solución del problema de localización del robot y aprendizaje del mapa simultáneamente, resueltos dentro del marco de las redes bayesianas por medio del uso de filtros de Partículas, pueden encontrarse en [Montemerlo2002] y [Montemerlo2003]. En [Zhou2003] se utiliza una red bayesiana (cuya estructura se aprende inicialmente) para planificar el movimiento de un robot con el objeto de que las medidas proporcionadas por el mismo permitan localizar al robot lo antes posible. En [Theocharous2004] se propone el uso de una red bayesiana multinivel para la construcción de mapas multicapa con los que simplificar el proceso de planificación del robot. Por último, en [Kawasaki2004] se propone una red bayesiana para construir la totalidad de la arquitectura de control del robot.

Además, es importante destacar el creciente número de publicaciones relacionadas con el uso del filtro de Partículas para los problemas de localización (a partir de una posición conocida, desconocida, o errónea) y aprendizaje del mapa (en sistemas con muchos elementos). Esto es

debido a su facilidad de implementación y a la capacidad de este tipo de algoritmos para utilizar distintos tipos de relaciones probabilísticas [Thrun2002].

Otras aplicaciones de las redes bayesianas como técnicas de fusión multisensorial son la validación de medidas sensoriales ([Ibarguengoytia1998]), la predicción de intenciones ([ZhangY2002], [Okello2003]), el reconocimiento y comprensión del lenguaje hablado ([Russell1999]), la detección de fallos y causas del incorrecto funcionamiento de sistemas ([Breese1996], [Betta2000], [Sochorova2000]), etc. Otras aplicaciones pueden encontrarse en [Lauritzen1999].

Todas estas aplicaciones ponen de manifiesto la utilidad de las redes bayesianas como técnicas de fusión, motivo por el que es la técnica elegida para unificar los dos subsistemas del sistema de fusión del robot autónomo móvil de la arquitectura de control presentada al final de la tesis. Aun más, las características de los algoritmos de inferencia exacta son el punto de partida del desarrollo de los algoritmos de fusión secuencial y asíncrona de medidas retardadas y temporalmente desordenadas que se proponen en los siguientes capítulos.

Capítulo II

Redes bayesianas para modelar sistemas lineales dinámicos con sensores asíncronos y medidas temporalmente retardadas

En este capítulo se realiza un análisis detallado del modelado con redes bayesianas del comportamiento de un *sistema lineal* de control sobre el que un conjunto de sensores realiza medidas de forma cíclica o acíclica. En el mismo, se lleva a cabo un estudio en profundidad del proceso de inferencia exacta sobre la red bayesiana elegida, dirigido a estimar el estado del sistema a partir de la información inicial, las señales de control y las medidas, cuyos valores podrán estar disponibles de forma temporalmente desordenada y no ser siempre válidos para estimar el estado del sistema.

El punto de partida de este estudio es la equivalencia existente entre filtro de Información y de Alisado, y el proceso de inferencia exacto sobre el árbol de unión asociado a la red bayesiana dinámica continua propuesta en [MurphyKP1998b]. Dicha red está formada en cada instante de tiempo t por dos nodos (uno asociado al estado del sistema y el otro a las medidas de un único sensor) y un enlace (que parte del nodo del estado y termina en el nodo de medida), y entre los instantes de tiempo $t-1$ y t existe un enlace adicional (que parte del nodo estado en el instante $t-1$ y termina en el nodo estado t). La equivalencia entre el filtro de Información y el proceso de inferencia exacto, demostrada en [MurphyKP1998b], se fundamenta en que la relación existente entre las variables de medida y de estado, y entre las variables de estado para dos instantes de muestreo sucesivos, se puede modelar por medio de funciones lineales que incluyen un término aditivo responsable de modelar un ruido blanco con una determinada covarianza. El tipo de funciones elegidas para modelar las relaciones probabilísticas entre las variables no viene determinado por la estructura de la red, sino por el hecho de que únicamente se pueden utilizar algoritmos de inferencia exacta para un número limitado de distribuciones, caracterizadas todas ellas por la existencia de expresiones integrales para las densidades condicionales elegidas. Esta limitación, que puede ser omitida utilizando algoritmos de inferencia aproximados, constituye en sí misma el soporte analítico de un detallado análisis del proceso de inferencia, por lo que durante este capítulo nos centraremos en el modelado del comportamiento de sistemas de control con relaciones lineales entre los estados y las medidas.

En este capítulo se extiende el análisis para el caso en el que el cambio de estado del sistema también dependa de una señal de control, y en vez de tener un único sensor tomando medidas periódicamente exista un conjunto de sensores que pueden proporcionar información cíclicamente, acíclicamente y/o con un retardo temporal, debido al tiempo consumido en el procesamiento o en la comunicación de la medida, no necesariamente constante. También se considera el caso en el que las medidas proporcionadas por los sensores no siempre son válidas y se modela, mediante la incorporación de nodos y enlaces adicionales, la existencia de una etapa de validación como parte de la estructura de la red.

El objetivo primordial de este capítulo es demostrar, para un sistema lineal dinámico con múltiples sensores, la equivalencia entre el filtro de Información distribuido y el proceso de inferencia exacto, basado en las operaciones asociadas a una etapa de recolección cuando se elige como nodo raíz al último estado, sobre el árbol de unión asociado a la red bayesiana continua que modela el comportamiento del sistema. En el filtro de Información distribuido, formado por etapas sucesivas de predicción del estado actual a partir del estado en el instante anterior y asimilación de la información sensorial del instante actual, las medidas, tanto síncronas como asíncronas, han de introducirse de forma temporalmente ordenada, en la etapa de asimilación correspondiente al instante de medida. Por lo tanto, el filtro de Información no permite tratar de forma natural la información sensorial que por motivos de procesamiento o retardos en el proceso de comunicación llega al mismo temporalmente retardada, después de que la etapa de asimilación de dicha medida ya ha sido superada. El proceso de inferencia exacto sobre la red bayesiana resuelve este problema de forma automática, ya que una de sus características principales es que el resultado de dicho proceso no depende del orden en el que se introduzca la evidencia en los nodos del árbol de unión, y por lo tanto es posible incorporar las medidas de forma temporalmente desordenada para el caso en el que existan retardos temporales en la incorporación de la información sensorial por parte del proceso de inferencia.

Durante la demostración de la equivalencia entre el filtro de Información y la etapa de recolección del proceso de inferencia exacta, se analizará la forma en la que el proceso de inferencia almacena la información sensorial, se observará como se incorpora y asimila la información que llega a la red temporalmente retardada, y se desarrollará un algoritmo de inferencia exacta que minimice, para la red bayesiana propuesta, el número de operaciones necesarias del proceso de inferencia general.

Este capítulo se encuentra organizado en dos secciones. En la primera, se introducen las características y operaciones que soportan el proceso de inferencia exacto sobre árboles de unión para redes bayesianas con variables vectoriales continuas y distribuciones gaussianas multivariable ([Lauritzen1992], [MurphyKP1998a]), y que son utilizadas durante las demostraciones de este capítulo. En la segunda, se demuestra la equivalencia entre el filtro de Información y el proceso de inferencia para la red que modela el comportamiento del sistema lineal de control con múltiples sensores (síncronos/asíncronos, que tienen un tiempo de procesamiento y comunicación variable, que puede introducir retardos en la recepción de la información sensorial por parte del algoritmo; y que no siempre producen medidas válidas). Dicha demostración se encuentra dividida en tres etapas diferenciadas. En la primera se presentan dos redes equivalentes para modelar la existencia de la señal de control, y se elige la más sencilla desde el punto de vista computacional para las etapas sucesivas. En la segunda se analiza exhaustivamente, para la red que modela el comportamiento del sistema multisensorial, el proceso de inferencia exacto, haciendo especial hincapié en las operaciones que se realizan cuando las medidas llegan temporalmente desordenadas, y se propone una primera versión del algoritmo de inferencia. Y en la tercera se modifica la red de la etapa anterior para que ésta incorpore, como parte del proceso de inferencia exacta, una etapa de validación, y se proponen nuevas versiones del algoritmo de inferencia.

C2.1. Inferencia exacta en redes bayesianas con variables continuas gaussianas

Las propiedades de las redes bayesianas no implican ninguna restricción sobre las características de las variables aleatorias almacenadas en las mismas o sobre las probabilidades condicionales que capturan sus relaciones. Por lo tanto, además de las redes bayesianas con variables discretas, también se pueden construir redes bayesianas continuas y mixtas, y en los tres casos utilizar las distribuciones condicionales que permitan construir un modelo probabilístico cercano a la realidad.

Los métodos de inferencia exactos utilizados para estimar el estado de conjuntos de variables de la red se fundamentan en la existencia de expresiones exactas para las integrales de las densidades condicionales elegidas [Cowell1998]. Debido a la imposibilidad de obtener dichas expresiones salvo para un número limitado de distribuciones, dichos métodos sólo pueden ser aplicados sobre un grupo restringido de redes. Sin embargo, este hecho no implica que no se pueda realizar inferencia sobre el resto de los casos, ya que existe un amplio y creciente grupo de algoritmos de inferencia aproximados. Así mismo, el origen de la limitación es en sí mismo un hecho ventajoso para el grupo de redes sobre los que se han desarrollado métodos de inferencia exactos, ya que la existencia de expresiones exactas durante todo el proceso permite realizar un estudio analítico del mismo.

En el caso de las redes bayesianas con variables continuas, la propagación exacta es factible cuando se utilizan variables gaussianas con relaciones de dependencia lineales. Para este tipo de redes se han propuesto métodos de inferencia exacta fundamentados en la eliminación de variables [Shachter1989] y en la propagación de la evidencia a través de los árboles de conglomerados asociados a la red ([Lauritzen1992] y [Lauritzen1999]). Los dos últimos métodos son también aplicables a redes bayesianas mixtas con distribuciones condicionales gaussianas multivariantes sobre las variables continuas, para las combinaciones de estados de las variables discretas de las que dependen. Además, al fundamentarse en la propagación de la evidencia, permiten realizar inferencias sobre todas las variables de la red bayesiana.

Las siguientes demostraciones se basan en el método más sencillo de los dos, el propuesto en [Lauritzen1992] que, a pesar de ser menos general que el de [Lauritzen1999] y de poder presentar problemas de estabilidad computacional, permite realizar un análisis completo del proceso de inferencia para las redes de este capítulo. Antes de comenzar con dicho análisis y con el objetivo de facilitar el posterior seguimiento de las demostraciones, en la siguiente sección se realiza una breve descripción del método de inferencia utilizado.

C2.1.1. Inferencia sobre árboles de unión para redes bayesianas con variables vectoriales continuas y distribuciones gaussianas multivariantes

En este capítulo se presenta una versión simplificada, para el caso de redes bayesianas con variables únicamente gaussianas, del método propuesto en [Lauritzen1992], y cuya implementación se detalla en [Olesen1993]. Esta descripción incluye las modificaciones presentadas en [MurphyKP1998a], que tienen como objetivo extender el método original que consideraba variables continuas unidimensionales al caso de variables continuas vectoriales. Además, el mecanismo de inferencia sobre el árbol de conglomerados propuesto en dicho método no es exactamente igual al algoritmo utilizado durante las demostraciones de este capítulo, en el que se realizarán las operaciones del algoritmo de Hugin ([JensenFV1990a], [JensenFV1990b]), en un orden diferente al preestablecido en dicho algoritmo.

El método de inferencia elegido se fundamenta en: (1) la construcción de un árbol cuyos nodos están compuestos por conglomerados (subconjuntos) de las variables de la red bayesiana original, y (2) en la realización de un conjunto de operaciones sobre las funciones de probabilidad conjunta, denominadas potenciales, asociadas a cada uno de los conglomerados del árbol.

El árbol de conglomerados (subconjuntos de variables de la red bayesiana) se construye de forma que se mantienen las relaciones de independencia existentes en la red original y que la estructura de dicho árbol soporte un método consistente de inferencia probabilística basado en el intercambio de mensajes. Para construir un árbol caracterizado por estas dos propiedades, se realizan los siguientes pasos (cuyo significado y ejecución se explican de forma detallada en [JensenFV1996], [Castillo1997] y [Cowell1999]):

- *Moralizar la red bayesiana*, añadiendo los enlaces no direccionales necesarios para que todos los padres de cada nodo de la red se encuentren unidos entre sí y *eliminando las direcciones de todos los enlaces* existentes.
- *Triangular el grafo no dirigido* obtenido en la etapa interior, añadiendo los enlaces no direccionales necesarios para asegurar que no hay ningún ciclo (cadena cerrada) que enlace más de tres nodos sin que exista ningún otro ciclo más corto (atajo) entre los mismos.
- *Identificar los conglomerados* del grafo triangulado como subconjuntos completos maximales de sus nodos, i.e. como subconjuntos de variables del grafo triangulado cuyos nodos se encuentran totalmente conectados entre sí y que no se encuentran a su vez estrictamente contenidos en otro subconjunto completo.
- *Construir el árbol de unión* asociado a la red bayesiana, utilizando un nodo por cada uno de los conglomerados identificados en la etapa anterior de forma que la estructura del árbol

cumpla la propiedad de intersección dinámica. Esta propiedad establece que toda variable V contenida en dos conglomerados conectados a través de un camino de conglomerados tiene que estar incluida en todos los conglomerados del camino.

Mientras la primera etapa de este proceso tiene como propósito asegurar que en el árbol de unión hay al menos un conglomerado en el que se encuentra cada variable de la red y sus padres (para poder asignarle la función de probabilidad condicional existente entre dichas variables), el objetivo de las siguientes es obtener una estructura que asegure las propiedades de independencia de la red original y que permita construir un algoritmo de inferencia consistente. El conjunto de enlaces añadidos en la etapa de triangulación no es único en la mayoría de los casos, por lo que de una misma red se pueden obtener distintos árboles de unión, todos ellos válidos. Además, se pueden añadir enlaces adicionales si se desea forzar la existencia de conglomerados con un conjunto determinado de variables. Sin embargo, la eficiencia computacional del algoritmo de inferencia está fuertemente ligada con la obtención de árboles de unión cuyos conglomerados tengan un espacio de estados pequeño, por lo que se han desarrollado múltiples algoritmos que optimizan el proceso de construcción del árbol de unión ([Kjaerulff1993], [JensenFV1994b], [Becker1996], [Olesen2002]).

Una vez construido el árbol de unión, a cada nodo (conglomerado) del mismo se le asigna *un potencial* $\Phi(V)$, *que es la función de probabilidad conjunta del conjunto de variables V contenido en su conglomerado correspondiente.* Realizando un conjunto de operaciones sobre dichos potenciales, se consigue transmitir información probabilística a través de los distintos nodos del árbol, y obtener la probabilidad asociada a distintas variables y/o subconjuntos de variables. Los diferentes algoritmos de inferencia desarrollados sobre el árbol de unión (Lauritzen-Spiegelhalter [Lauritzen1988], Hugin [JensenFV1990a, JensenFV1990b] y Shafer-Shenoy [Shafer1990b]) se diferencian en la forma y orden en la que se realizan las operaciones (una comparación detallada de los tres algoritmos puede verse en [Lepar1998]). En nuestro caso, utilizaremos una arquitectura que se fundamenta en el algoritmo de Hugin a la hora de almacenar los potenciales y realizar las operaciones, pero que se diferencia del mismo en el orden en el que éstas se realizan.

Para realizar las operaciones del algoritmo de Hugin es necesario: (1) *añadir un nodo auxiliar, llamado separador, sobre los enlaces que unen los diferentes nodos del árbol;* (2) *asignar a cada separador el conjunto de variables resultante de la intersección de las variables de los dos nodos de conglomerados que enlaza;* y (3) *almacenar una función potencial de probabilidad conjunta para cada conglomerado y cada separador.*

Tras el proceso de inicialización de los potenciales, la probabilidad conjunta sobre cada conglomerado se obtiene tras llevar a cabo, de forma sucesiva y ordenada, operaciones de absorción sobre parejas de conglomerados adyacentes.

La operación de absorción definida en esta arquitectura es el resultado de realizar, sobre los potenciales $\Phi(V)$, $\Phi(W)$ y $\Phi(S)$, asociados a cada par de conglomerados adyacentes W y V que se encuentran conectados a través de un separador S , las operaciones recogidas en las expresiones (C2-1), (C2-2), (C2-3) y (C2-4). En la figura siguiente se esquematiza la relación existente entre los conglomerados y separadores relacionados por la operación de absorción recogida en dichas expresiones.

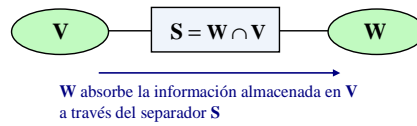


Figura C2-1. Operación de absorción entre conglomerados adyacentes.

$$\Phi^*(S) = \int \Phi(V) dZ \Big|_{Z=V \setminus S} \quad (C2.1)$$

$$\Phi^*(W) = \Phi(W) \frac{\Phi^*(S)}{\Phi(S)} \quad (C2.2)$$

$$\Phi(W) = \Phi^*(W) \quad (C2.3)$$

$$\Phi(S) = \Phi^*(S) \quad (C2.4)$$

Tras ejecutar estas operaciones se dice que el potencial del conglomerado W absorbe la información probabilística almacenada en el potencial del conglomerado V . La expresión (C2-1) se utiliza para calcular el nuevo valor del potencial del separador, como una marginalización del potencial del conglomerado V sobre las variables del mismo que no se encuentren en el separador, mientras que la ecuación (C2-2) se utiliza para calcular el potencial del conglomerado W . Las fórmulas (C2-3) y (C2-4) son las actualizaciones de los potenciales del conglomerado W y el separador S .

El orden establecido en la arquitectura de Hugin se basa en la elección de un conglomerado del árbol como raíz y en la división del proceso recursivo de absorción en dos etapas: una, llamada *recolección*, en la que se recoge la evidencia desde los conglomerados hoja hacia el raíz y otra, denominada *distribución*, en la que se distribuye la evidencia recogida desde el conglomerado raíz hacia los hoja. Cuando el proceso se realiza sin introducir ningún tipo de evidencia (información sobre el estado de las variables) el potencial almacenado en cada conglomerado y separador contiene la probabilidad a priori conjunta del conjunto de variables que almacena. Una vez introducida la evidencia en el árbol de unión y realizadas las dos etapas del proceso de absorción,

los potenciales asociados a cada conglomerado y separador almacenan la probabilidad conjunta de sus variables y la evidencia aportada.

La probabilidad conjunta de todas las variables de la red y la evidencia incorporada se puede calcular con la expresión (C2-5), donde \mathbf{e} representa la evidencia introducida, \mathbf{X}_C los conjuntos de variables de los conglomerados de la red, \mathbf{X}_S los conjuntos de variables de los separadores y \mathbf{X} el conjunto de todas las variables de la red bayesiana:

$$\Phi(\mathbf{X}, \mathbf{e}) = \frac{\prod_C \Phi(\mathbf{X}_C, \mathbf{e})}{\prod_S \Phi(\mathbf{X}_S, \mathbf{e})} \quad (\text{C2.5})$$

El método presentado para la construcción del árbol de conglomerados y para el algoritmo de inferencia puede ser utilizado para redes con variables discretas, continuas (con distribuciones gaussianas multivariable) o mixtas. Sin embargo, al utilizarse potenciales distintos para cada caso, las operaciones sobre los mismos son diferentes.

Las operaciones sobre las que se sustenta el algoritmo de Hugin para el caso puramente continuo con distribuciones gaussianas multivariables con dependencias puramente lineales, que se recogen en la tabla C2-2, se realizan sobre funciones de probabilidad conjunta gaussiana multivariable. Aunque la representación habitual de estas funciones suele ser bajo su *forma normal*, las operaciones propuestas para el algoritmo de Hugin son más sencillas cuando se utiliza la *representación canónica*. En la siguiente tabla se recogen ambas representaciones, y la equivalencia existente entre las mismas.

<p style="text-align: center;">REPRESENTACION NORMAL</p> $\Pr(\mathbf{x}) = N_{\mathbf{x}}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}) = p e^{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}})}$	$g = \log p - \frac{1}{2} \boldsymbol{\mu}_{\mathbf{x}}^T \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \boldsymbol{\mu}_{\mathbf{x}}$ $\mathbf{h} = \boldsymbol{\Sigma}_{\mathbf{x}}^{-1} \boldsymbol{\mu}_{\mathbf{x}}$ $\mathbf{K} = \boldsymbol{\Sigma}_{\mathbf{x}}^{-1}$
<p style="text-align: center;">REPRESENTACION CANONICA</p> $\Phi(\mathbf{x}) = (g, \mathbf{h}, \mathbf{K}) = e^{g + \mathbf{x}^T \mathbf{h} - \frac{1}{2} \mathbf{x}^T \mathbf{K} \mathbf{x}}$	

Tabla C2-1. Representaciones de una función de probabilidad conjunta y la relación existente entre ellas.

En la tabla siguiente se presentan las operaciones necesarias para realizar las etapas de inicialización, absorción e incorporación de la evidencia, en la versión extendida propuesta en [MurphyKP1998a] para el caso de redes bayesianas con variables continuas vectoriales. En dichas operaciones g_i son valores escalares, \mathbf{x}_i , \mathbf{y} , $\boldsymbol{\mu}_y$, $\boldsymbol{\Psi}$ y \mathbf{h}_i son vectores, y \mathbf{B}_i , $\boldsymbol{\Sigma}_y$ y \mathbf{K}_i son matrices. Además, para facilitar la comprensión de las demostraciones de los próximos apartados, se utilizará un color distinto para representar cada elemento de la tripleta $(g, \mathbf{h}, \mathbf{K})$ de la forma canónica y se obviará el cálculo del elemento g .

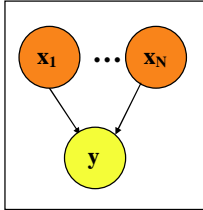
Inicialización: Conversión de forma Normal a Canónica	
	$\Pr(\mathbf{y} \mathbf{x}_1, \dots, \mathbf{x}_N) = \mathcal{N}_y(\boldsymbol{\mu}_y + \sum_{i=1..N} \mathbf{B}_i \mathbf{x}_i, \boldsymbol{\Sigma}_y) =$ $= (2\pi)^{-p/2} \boldsymbol{\Sigma}_y ^{-1/2} e^{-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_y - \sum_{i=1..N} \mathbf{B}_i \mathbf{x}_i)^T \boldsymbol{\Sigma}_y^{-1} (\mathbf{y} - \boldsymbol{\mu}_y - \sum_{i=1..N} \mathbf{B}_i \mathbf{x}_i)}$ $\Rightarrow \Phi_c(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{y}) = \left(-\frac{1}{2} \boldsymbol{\mu}_y^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{\mu}_y - \frac{p}{2} \log(2\pi) - \frac{1}{2} \log \boldsymbol{\Sigma}_y , \begin{bmatrix} -\mathbf{B}_1^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{\mu}_y \\ \vdots \\ -\mathbf{B}_N^T \boldsymbol{\Sigma}_y^{-1} \boldsymbol{\mu}_y \\ \boldsymbol{\Sigma}_y^{-1} \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \mathbf{B}_1^T \boldsymbol{\Sigma}_y^{-1} \mathbf{B}_1 & \dots & \mathbf{B}_1^T \boldsymbol{\Sigma}_y^{-1} \mathbf{B}_N & -\mathbf{B}_1^T \boldsymbol{\Sigma}_y^{-1} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{B}_N^T \boldsymbol{\Sigma}_y^{-1} \mathbf{B}_1 & \dots & \mathbf{B}_N^T \boldsymbol{\Sigma}_y^{-1} \mathbf{B}_N & -\mathbf{B}_N^T \boldsymbol{\Sigma}_y^{-1} \\ -\boldsymbol{\Sigma}_y^{-1} \mathbf{B}_1 & \dots & -\boldsymbol{\Sigma}_y^{-1} \mathbf{B}_N & \boldsymbol{\Sigma}_y^{-1} \end{bmatrix} \right)$ <p>siendo p la dimension de la variable aleatoria vectorial \mathbf{y}</p>
Extensión	$\Phi(\mathbf{x}) = (\mathbf{g}, \mathbf{h}, \mathbf{K}) \Rightarrow \Phi(\mathbf{x}, \mathbf{y}) = \left(\mathbf{g}, \begin{bmatrix} \mathbf{h} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right)$
Reorganización	$\Phi(\mathbf{x}, \mathbf{y}) = \left(\mathbf{g}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_y \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix} \right) \Leftrightarrow \Phi(\mathbf{y}, \mathbf{x}) = \left(\mathbf{g}, \begin{bmatrix} \mathbf{h}_y \\ \mathbf{h}_x \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{yy} & \mathbf{K}_{yx} \\ \mathbf{K}_{xy} & \mathbf{K}_{xx} \end{bmatrix} \right)$
Multiplicación	$\left. \begin{array}{l} \Phi_1(\mathbf{x}) = (\mathbf{g}_1, \mathbf{h}_1, \mathbf{K}_1) \\ \Phi_2(\mathbf{x}) = (\mathbf{g}_2, \mathbf{h}_2, \mathbf{K}_2) \end{array} \right\} \Rightarrow \Phi_1(\mathbf{x}) \cdot \Phi_2(\mathbf{x}) = (\mathbf{g}_1 + \mathbf{g}_2, \mathbf{h}_1 + \mathbf{h}_2, \mathbf{K}_1 + \mathbf{K}_2)$
División	$\left. \begin{array}{l} \Phi_1(\mathbf{x}) = (\mathbf{g}_1, \mathbf{h}_1, \mathbf{K}_1) \\ \Phi_2(\mathbf{x}) = (\mathbf{g}_2, \mathbf{h}_2, \mathbf{K}_2) \end{array} \right\} \Rightarrow \frac{\Phi_1(\mathbf{x})}{\Phi_2(\mathbf{x})} = (\mathbf{g}_1 - \mathbf{g}_2, \mathbf{h}_1 - \mathbf{h}_2, \mathbf{K}_1 - \mathbf{K}_2)$
Marginalización	$\Phi(\mathbf{x}, \mathbf{y}) = \left(\mathbf{g}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_y \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix} \right) \Rightarrow \Phi^*(\mathbf{y}) = \int \Phi(\mathbf{x}, \mathbf{y}) d\mathbf{x} \Rightarrow$ $\Phi^*(\mathbf{y}) = \left(\mathbf{g} + \frac{p}{2} \log(2\pi) - \log \mathbf{K}_{xx} + \mathbf{h}_x^T \mathbf{K}_{xx}^{-1} \mathbf{h}_x, \begin{bmatrix} \mathbf{h}_y - \mathbf{K}_{yx} \mathbf{K}_{xx}^{-1} \mathbf{h}_x \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{yy} - \mathbf{K}_{yx} \mathbf{K}_{xx}^{-1} \mathbf{K}_{xy} \end{bmatrix} \right)$ <p>siendo p la dimension de la variable aleatoria vectorial \mathbf{y}</p>
Incorporar Evidencia	<p>Dado $\Phi(\mathbf{x}, \mathbf{y}) = \left(\mathbf{g}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_y \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix} \right)$ y la evidencia en $\mathbf{y} = \boldsymbol{\psi} \Rightarrow$</p> $\Phi(\mathbf{x}, \boldsymbol{\psi}) = \Phi(\mathbf{x}, \mathbf{y} _{\mathbf{y}=\boldsymbol{\psi}}) = \left(\mathbf{g} + \mathbf{h}_y^T \boldsymbol{\psi} - \frac{1}{2} \boldsymbol{\psi}^T \mathbf{K}_{yy} \boldsymbol{\psi}, \begin{bmatrix} \mathbf{h}_x - \mathbf{K}_{xy} \boldsymbol{\psi} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{xx} \end{bmatrix} \right)$

Tabla C2-2. Operaciones sobre potenciales canónicos para redes puramente continuas con distribuciones gaussianas multivariable.

La operación de inicialización se utiliza para realizar la conversión de una función de probabilidad gaussiana multivariable con dependencias puramente lineales en forma normal a su expresión en forma canónica. Es decir, para obtener los valores de la tripleta $(\mathbf{g}, \mathbf{h}, \mathbf{K})$ correspondientes a los valores de $\boldsymbol{\mu}_y$, \mathbf{B}_i y $\boldsymbol{\Sigma}_y$. Como la información probabilística asociada a la función en ambas formas es equivalente, el resultado de dicha operación se almacena en el potencial condicional $\Phi_c(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{y})$, en el que la última variable (\mathbf{y}) depende condicionalmente de las anteriores $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$.

Las operaciones de extensión y reorganización tienen como objeto ampliar el número de variables de un potencial o reorganizar el orden en las que las variables aparecen, sin modificar la información probabilística almacenada en los mismos. Las operaciones de marginalización, multiplicación y división son utilizadas en las expresiones (C2-1) y (C2-2) de la operación de absorción. La operación de incorporar evidencia es utilizada para introducir en los potenciales de la red el conocimiento que se tiene sobre el valor de una determinada variable.

Para este tipo de redes los potenciales de los conglomerados se inicializan como el producto de los potenciales condicionales canónicos correspondientes a las funciones de probabilidad condicional de los nodos que contienen, extendidas y reorganizadas sobre el conjunto de todas las variables del conglomerado. Es decir, mediante la expresión de inicialización de la tabla C2-2 se extraen los valores de la tripleta (g, h, K) asociados a cada potencial condicional Φ_c de la red bayesiana, y para cada conglomerado, los potenciales Φ (de probabilidad conjunta) de los conglomerados se obtienen como el producto de los potenciales condicionales de las variables que contiene, tras realizar las operaciones de extensión y reorganización necesarias. Además, durante el proceso de inicialización, los tres campos de los potenciales de los separadores se igualan a $\mathbf{0}$ (escalar, vectorial y matricial respectivamente), de forma que almacenen información probabilística nula.

Es importante destacar en este punto, la diferente notación utilizada para representar la función de probabilidad condicional en forma canónica, obtenida como conversión de la expresión en forma normal, y denominada potencial condicional, y la función de probabilidad conjunta, obtenida tras el proceso total de inicialización y actualizada tras procesos de recolección y distribución, denominada únicamente potencial. El potencial condicional lo representaremos como $\Phi_c(\mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{y})$, donde la última variable (\mathbf{y}) depende de las anteriores ($\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$), mientras que el potencial (conjunto) lo representaremos mediante $\Phi(\mathbf{V})$, con \mathbf{V} un conjunto de variables cualquiera.

A la hora de incorporar la evidencia (el conocimiento del estado) de una de las variables de la red, se realiza la última operación de la tabla C2-2 sobre todos los potenciales que contienen a dicha variable.

Por último es importante destacar que el orden preestablecido en el proceso de propagación de Hugin (con un nodo raíz que dirige el orden de recolección-distribución) no es realmente necesario, ya que la realización de operaciones de absorción en orden aleatorio también permite que se alcance el estado de equilibrio final, siempre y cuando se propague toda la información a priori y evidencia a través de todos los conglomerados y separadores. El principal inconveniente de un proceso totalmente aleatorio es la realización de un número elevado de absorciones que modificarán el estado de los potenciales únicamente con una parte de la evidencia incorporada en la red, y en el peor de los casos, la realización de absorciones que no produzcan ninguna

modificación. Además, hay que tener en cuenta que se obtienen los mismos resultados introduciendo toda la evidencia simultáneamente y realizando a continuación el proceso de recolección-distribución, que cuando se introduce en un orden arbitrario y en etapas sucesivas cada elemento de evidencia por separado, y se realiza tras cada incorporación de evidencia un proceso completo de recolección-distribución. La posibilidad de no utilizar un orden preestablecido y de poder incorporar la evidencia por partes en un orden arbitrario se fundamenta en la existencia de los separadores que almacenan los potenciales previamente propagados y en la ecuación (C2-2), donde el potencial del conglomerado se actualiza añadiéndole y quitándole, respectivamente, el nuevo y viejo valor del potencial del separador.

C2.2. Redes bayesianas continuas y fusión multisensorial, asíncrona y temporalmente desordenada

En este apartado se hace un estudio detallado del proceso de inferencia sobre una red bayesiana que modela el comportamiento de un sistema lineal dinámico y el de un conjunto formado por múltiples sensores, que proporcionan información del sistema de forma asíncrona y temporalmente desordenada.

El análisis del proceso de inferencia sobre la red bayesiana de la figura C2-2, que modela el comportamiento de un sistema lineal y el de un único sensor que toma medidas del sistema periódicamente, puede encontrarse en [MurphyKP1998b]. La conclusión de dicho estudio es que cuando se elige como nodo raíz el último estado \mathbf{x}_n , el resultado del proceso de inferencia después de la etapa de recolección es equivalente al del filtro de Información, y después de la etapa de distribución es igual al del de alisado de Kalman. Las expresiones para el filtro y el alisado se pueden encontrar en [Bar-Shalom1993]. Las del filtro también se presentan en el apéndice I.

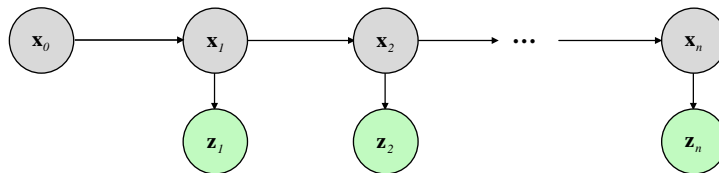


Figura C2-2. Red bayesiana que modela el comportamiento de un sistema lineal dinámico (representado por la variable \mathbf{x}) con un único sensor que proporciona cíclicamente la medida \mathbf{z}_i .

El estudio de este capítulo se centra en la etapa de recolección, extendiendo el análisis de [MurphyKP1998b] para el caso en el que el cambio de estado del sistema lineal también dependa de una señal de control, y en vez de tener un único sensor tomando medidas periódicamente exista un conjunto de S sensores que pueden proporcionar información cíclicamente, acíclicamente y/o con

un retardo temporal no necesariamente constante. El comportamiento de un sistema de estas características, puede ser modelado por la red bayesiana de la siguiente figura.

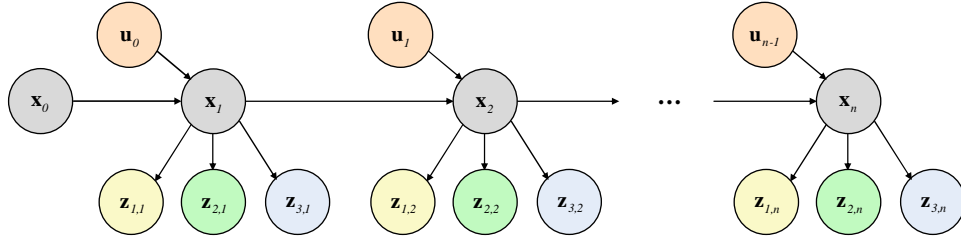


Figura C2-3. Red bayesiana que modela el comportamiento de un sistema lineal dinámico, cuyo estado \mathbf{x}_t depende del estado \mathbf{x}_{t-1} en el instante anterior y de la señal de control \mathbf{u}_{t-1} , y sobre el que $S=3$ sensores pueden proporcionar medidas $\mathbf{z}_{s,t}$ de forma acíclica y temporalmente retardada.

En la red de la figura C2-3 se considera la existencia de tres sensores y se incluyen nodos sensoriales que permiten introducir las medidas tomadas de los tres sensores en todos los instantes de muestreo. La existencia de tres nodos sensoriales para todos los instantes de muestreo no implica la necesidad de que se introduzca evidencia para todos los sensores para todos los instantes, si no simplemente la posibilidad de incorporarla en caso de que ésta existiese. Además, el tratamiento de la información sensorial que llega retardada respecto al instante de la última estima del estado no es diferente al tratamiento del resto de la información sensorial debido a las propiedades del algoritmo de propagación exacto de inferencia basado en el árbol de unión. Como ya se apuntó, la evidencia (información sensorial para nuestra red) puede ser introducida por partes, cuando se encuentre disponible, y en cualquier orden sin modificar el resultado final del proceso de inferencia.

El análisis de las expresiones exactas del proceso de inferencia se encuentra dividido en las siguientes secciones. Inicialmente, se presentan dos redes distintas que incluyen la señal de control en el modelo del sistema y se demuestra su equivalencia analíticamente. A continuación se estudia el comportamiento del proceso de inferencia desde el punto de vista sensorial para el caso en el que existen múltiples sensores funcionando de forma síncrona, asíncrona y/o con retardos temporales. En dicho apartado, además se determina el número mínimo de absorciones para que el funcionamiento del algoritmo de inferencia sea el mismo que el de un filtro de Información. Por último, se estudia la posibilidad de incorporar una etapa de validación antes de introducir las medidas sensoriales y los problemas que dicha etapa conlleva en el caso de la existencia de medidas que llegan a la red bayesiana desordenadas temporalmente.

C2.2.1. Redes Bayesianas equivalentes para sistemas lineales de control

En esta sección se proponen dos redes bayesianas equivalentes para modelar un sistema de control lineal discreto cuyo estado \mathbf{x}_{t+1} en el instante $t+1$: (1) depende del estado \mathbf{x}_t en el instante t y de la señal de control \mathbf{u}_t aplicada durante el periodo de muestreo, y (2) es observado por un sensor que proporciona en ese instante la medida \mathbf{z}_{t+1} . Los modelos del sistema (C2-6) y sensorial (C2-7) son funciones lineales no deterministas, que incluyen un término aditivo responsable de modelar ruido blanco (\mathbf{v}_t y \mathbf{v}_{t+1}) con sus respectivas covarianzas (\mathbf{Q}_t y \mathbf{R}_{t+1}).

$$\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{u}_t + \mathbf{v}_t \quad (\text{C2.6})$$

$$\mathbf{z}_{t+1} = \mathbf{H}_{t+1} \mathbf{x}_{t+1} + \mathbf{v}_{t+1} \quad (\text{C2.7})$$

El objetivo del problema es estimar el estado del sistema y su covarianza a partir de un estado y covarianza iniciales, las ordenes de control utilizadas y la información sensorial disponible.

Las dos redes bayesianas de la siguiente tabla permiten modelar y resolver este problema de forma equivalente.

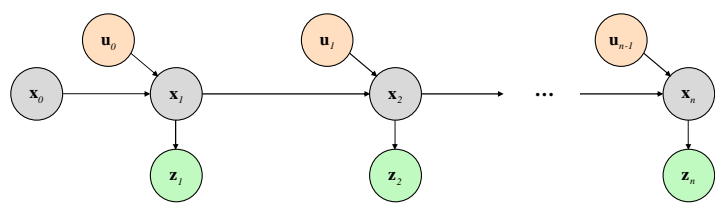
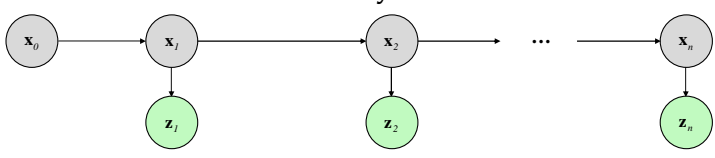
<p style="text-align: center;">Red Bayesiana (a)</p>  <p style="text-align: center;"> $\Pr(\mathbf{x}_0) = N_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$ $\Pr(\mathbf{u}_t) = N_{\mathbf{u}_t}(\hat{\mathbf{u}}_t, \mathbf{U}_t)$ $\Pr(\mathbf{x}_{t+1} \mathbf{x}_t, \mathbf{u}_t) = N_{\mathbf{x}_{t+1}}(\mathbf{0} + \mathbf{F}_t \mathbf{x}_t + \mathbf{1} \mathbf{u}_t, \mathbf{Q}_t(a))$ $\Pr(\mathbf{z}_t \mathbf{x}_t) = N_{\mathbf{z}_t}(\mathbf{0} + \mathbf{H}_t \mathbf{x}_t, \mathbf{R}_t)$ </p>	
<p style="text-align: center;">Red Bayesiana (b)</p>  <p style="text-align: center;"> $\Pr(\mathbf{x}_0) = N_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$ $\Pr(\mathbf{x}_{t+1} \mathbf{x}_t) = N_{\mathbf{x}_{t+1}}(\mathbf{u}_t + \mathbf{F}_t \mathbf{x}_t, \mathbf{Q}_t(b))$ $\Pr(\mathbf{z}_t \mathbf{x}_t) = N_{\mathbf{z}_t}(\mathbf{0} + \mathbf{H}_t \mathbf{x}_t, \mathbf{R}_t)$ </p>	
<p>El objetivo en ambos casos es obtener :</p> <p style="text-align: center;"> $\hat{\mathbf{X}}_n \setminus \{\hat{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{z}_t = \zeta_t, \mathbf{H}_t, \mathbf{R}_t, \mathbf{u}_{t-1}, \mathbf{F}_{t-1}; t=1:n\}$ y $\mathbf{P}_n \setminus \{\hat{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{z}_t = \zeta_t, \mathbf{H}_t, \mathbf{R}_t, \mathbf{u}_{t-1}, \mathbf{F}_{t-1}; t=1:n\}$ </p>	

Tabla C2-3. Redes bayesianas equivalentes para modelar un sistema lineal dependiente del estado anterior y de una señal de control, sobre el que un único sensor puede realizar medidas de forma acíclica y temporalmente retardada.

En el primer caso, la red bayesiana (a), la señal de control es modelada como una variable gaussiana y la dependencia entre el estado y dicha señal de control es capturada por la estructura de la red y la función de probabilidad condicional $\Pr(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{u}_t) = N_{\mathbf{x}_{t+1}}(\mathbf{0} + \mathbf{1} \mathbf{u}_t + \mathbf{F}_t \mathbf{x}_t, \mathbf{Q}_t(a))$. En esta función se considera que la variable \mathbf{x}_{t+1} es una variable aleatoria gaussiana, con media $\mathbf{0}$ y covarianza $\mathbf{Q}_t(a)$,

que depende linealmente de las variables gaussianas \mathbf{x}_t y \mathbf{u}_t . Además, para cada instante de muestreo se considera que hay una señal de control cuyo valor medio y covarianza es $\hat{\mathbf{u}}_t$ y $\hat{\mathbf{U}}_t$ respectivamente. En el segundo caso, la red bayesiana (b), no se utiliza una variable aleatoria para modelar la señal de control sino que dicha señal se incluye en la ecuación de probabilidad condicional $\Pr(\mathbf{x}_{t+1} | \mathbf{x}_t) = \mathcal{N}_{\mathbf{x}_{t+1}}(\mathbf{u}_t + \mathbf{F}_t \mathbf{x}_t, \mathbf{Q}_t(b))$ como el valor medio correspondiente a la variable \mathbf{x}_{t+1} . Dicho de otra forma, se considera que la variable \mathbf{x}_{t+1} es una variable aleatoria gaussiana con media \mathbf{u}_t , covarianza $\mathbf{Q}_t(b)$ y sólo linealmente dependiente de la variable gaussiana \mathbf{x}_t .

Para demostrar la equivalencia entre las dos redes y familiarizar al lector con el proceso de inferencia construiremos el árbol de unión asociado a cada red y obtendremos los potenciales asociados a los conglomerados y separadores tras el proceso de inicialización.

La construcción de los árboles de unión en estos dos casos es muy sencilla, ya que después de la etapa de moralización se obtiene directamente un grafo ya triangulado muy sencillo, en el que es directo identificar los nodos asociados a cada conglomerado y construir un árbol de conglomerados que cumpla la propiedad de intersección dinámica. Los árboles de unión obtenidos para las redes bayesianas (a) y (b) se encuentran en la parte superior de las tablas C2-4 y C2-5 respectivamente. En ambos casos se han representado las variables asociadas a cada conglomerado (nodos elípticos) y a los separadores (nodos rectangulares). La única diferencia existente entre ambos árboles de unión es que el de la red bayesiana (a) incluye la variable de control en los conglomerados asociados a la transición $\mathbf{x}_t \mathbf{u}_t \mathbf{x}_{t+1}$ mientras que en el caso de la red bayesiana (b), cuyo conglomerado de transición está formado por las variables $\mathbf{x}_t \mathbf{x}_{t+1}$, dicha variable no aparece.

Para inicializar los potenciales asociados a los conglomerados hay que convertir las funciones de probabilidad condicional propuestas en el planteamiento del problema, que se encuentran en forma normal, en potenciales condicionales canónicos Φ_c , utilizando para dicha conversión la operación de inicialización de la tabla C2-2. Los potenciales obtenidos después de la conversión para cada una de las redes se encuentran en la parte central de las tablas C2-4 y C2-5. Las variables $\hat{\mathbf{y}}_0$ e \mathbf{Y}_0 que aparecen en ambos casos tras la realización de la conversión en la que se obtiene el potencial condicional asociado al valor inicial del estado, $\Phi_c(\mathbf{x}_0) = (\mathcal{G}, \mathbf{P}_0^{-1} \hat{\mathbf{x}}_0, \mathbf{P}_0^{-1}) = (\mathcal{G}, \hat{\mathbf{y}}_0, \mathbf{Y}_0)$, no son otras que las variables de información sobre el estado, y de la covarianza de la información sobre el estado, utilizadas en las expresiones del filtro de Información. En el apéndice I se analiza la relación existente entre el espacio de estados (determinado por las variables $\hat{\mathbf{x}}_0$ y \mathbf{P}_0) y el espacio de información sobre el estado (determinado por las variables $\hat{\mathbf{y}}_0$ e \mathbf{Y}_0).

Los potenciales asociados a cada conglomerado se inicializan como el producto de los potenciales condicionales canónicos Φ_c en los que aparecen subconjuntos de los nodos del conglomerado, reorganizados y ampliados en el caso que sea necesario, de acuerdo con las operaciones recogidas

en la tabla C2-2. Los tres campos de los potenciales asociados a los separadores se inicializan a 0 (escalar, vectorial y matricial respectivamente). El proceso de inicialización, en el que se especifica el potencial condicional canónico asociado a cada conglomerado se encuentra recogido, para las redes bayesianas (a) y (b), en la parte inferior de las tablas C2-4 y C2-5.

Árbol de conglomerados del problema (a)	
Conversión de los potenciales normales en canónicos	
$\Pr(\mathbf{x}_0) = N_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, \mathbf{P}_0) \Rightarrow \Phi_c(\mathbf{x}_0) = (\mathbf{g}, \mathbf{P}_0^{-1}\hat{\mathbf{x}}_0, \mathbf{P}_0^{-1}) = (\mathbf{g}, \hat{\mathbf{y}}_0, \mathbf{Y}_0)$ $\Pr(\mathbf{u}_t) = N_{\mathbf{u}_t}(\hat{\mathbf{u}}_t, \mathbf{U}_0) \Rightarrow \Phi_c(\mathbf{u}_t) = (\mathbf{g}, \mathbf{U}_t^{-1}\hat{\mathbf{u}}_t, \mathbf{U}_t^{-1})$ $\Pr(\mathbf{x}_{t+1} \mathbf{x}_t, \mathbf{u}_t) = N_{\mathbf{x}_{t+1}}(\mathbf{0} + \mathbf{1}\mathbf{u}_t + \mathbf{F}_t\mathbf{x}_t, \mathbf{Q}_t(a)) \Rightarrow$ $\Phi_c(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) = \left(\mathbf{g}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} & -\mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \\ (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(a))^{-1} & -(\mathbf{Q}_t(a))^{-1} \\ -(\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & -(\mathbf{Q}_t(a))^{-1} & (\mathbf{Q}_t(a))^{-1} \end{bmatrix} \right)$ $\Pr(\mathbf{z}_t \mathbf{x}_t) = N_{\mathbf{z}_t}(\mathbf{0} + \mathbf{H}_t\mathbf{x}_t, \mathbf{R}_t) \Rightarrow \Phi_c(\mathbf{x}_t, \mathbf{z}_t) = \left(\mathbf{g}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_t^T \mathbf{R}_t^{-1} \mathbf{H}_t & -\mathbf{H}_t^T \mathbf{R}_t^{-1} \\ -\mathbf{R}_t^{-1} \mathbf{H}_t & \mathbf{R}_t^{-1} \end{bmatrix} \right)$	
Inicialización de los potenciales de los conglomerados	
(1)	$\Phi_{\mathbf{x}_0\mathbf{u}_0\mathbf{x}_1}^0(\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1) = \Phi_c(\mathbf{x}_0) \cdot \Phi_c(\mathbf{u}_0) \cdot \Phi_c(\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1) =$ $= \left(\mathbf{g}, \begin{bmatrix} \hat{\mathbf{y}}_0 \\ \mathbf{U}_0^{-1}\hat{\mathbf{u}}_0 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_0 + \mathbf{F}_0^T (\mathbf{Q}_0(a))^{-1} \mathbf{F}_0 & \mathbf{F}_0^T (\mathbf{Q}_0(a))^{-1} & -\mathbf{F}_0^T (\mathbf{Q}_0(a))^{-1} \\ (\mathbf{Q}_0(a))^{-1} \mathbf{F}_0 & (\mathbf{Q}_0(a))^{-1} + \mathbf{U}_0^{-1} & -(\mathbf{Q}_0(a))^{-1} \\ -(\mathbf{Q}_0(a))^{-1} \mathbf{F}_0 & -(\mathbf{Q}_0(a))^{-1} & (\mathbf{Q}_0(a))^{-1} \end{bmatrix} \right)$
(2)	$\Phi_{\mathbf{x}_t\mathbf{u}_t\mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) = \Phi_c(\mathbf{u}_t) \cdot \Phi_c(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) =$ $= \left(\mathbf{g}, \begin{bmatrix} \mathbf{0} \\ \mathbf{U}_t^{-1}\hat{\mathbf{u}}_t \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} & -\mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \\ (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(a))^{-1} + \mathbf{U}_t^{-1} & -(\mathbf{Q}_t(a))^{-1} \\ -(\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & -(\mathbf{Q}_t(a))^{-1} & (\mathbf{Q}_t(a))^{-1} \end{bmatrix} \right) \text{ con } t = 1 : n-1$
(3)	$\Phi_{\mathbf{x}_t\mathbf{z}_t}^0(\mathbf{x}_t, \mathbf{z}_t) = \Phi_c(\mathbf{x}_t, \mathbf{z}_t) = \left(\mathbf{g}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_t^T \mathbf{R}_t^{-1} \mathbf{H}_t & -\mathbf{H}_t^T \mathbf{R}_t^{-1} \\ -\mathbf{R}_t^{-1} \mathbf{H}_t & \mathbf{R}_t^{-1} \end{bmatrix} \right) \text{ con } t = 1 : n$
(4)	$\Phi_{\mathbf{x}_t\mathbf{z}_t \cap \mathbf{x}_{t-1}\mathbf{x}_t}^0(\mathbf{x}_t) = (\mathbf{0}, [\mathbf{0}], [\mathbf{0}]) \text{ con } t = 1 : n$
(5)	$\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}^0(\mathbf{x}_t) = (\mathbf{0}, [\mathbf{0}], [\mathbf{0}]) \text{ con } t = 1 : n-1$

Tabla C2-4. Árbol de unión asociado a la red bayesiana (a) y los valores de los potenciales de sus conglomerados (1,2 y 3) y separadores (4 y 5) tras el proceso de inicialización.

Árbol de conglomerados del problema (b)	
Conversión de los potenciales normales en canónicos	
$\Pr(\mathbf{x}_0) = N_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, \mathbf{P}_0) \Rightarrow \Phi_c(\mathbf{x}_0) = (g, \mathbf{P}_0^{-1}\hat{\mathbf{x}}_0, \mathbf{P}_0^{-1}) = (g, \hat{\mathbf{y}}_0, \mathbf{Y}_0)$ $\Pr(\mathbf{x}_{t+1} \mathbf{x}_t) = N_{\mathbf{x}_{t+1}}(\mathbf{u}_t + \mathbf{F}_t \mathbf{x}_t, \mathbf{Q}_t(\mathbf{b})) \Rightarrow$ $\Phi_c(\mathbf{x}_t, \mathbf{x}_{t+1}) = \left(g, \begin{bmatrix} -\mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{b}))^{-1} \mathbf{u}_t \\ (\mathbf{Q}_t(\mathbf{b}))^{-1} \mathbf{u}_t \end{bmatrix}, \begin{bmatrix} \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{b}))^{-1} \mathbf{F}_t & -\mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{b}))^{-1} \\ -(\mathbf{Q}_t(\mathbf{b}))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(\mathbf{b}))^{-1} \end{bmatrix} \right)$ $\Pr(\mathbf{z}_t \mathbf{x}_t) = N_{\mathbf{z}_t}(\mathbf{0} + \mathbf{H}_t \mathbf{x}_t, \mathbf{R}_t) \Rightarrow \Phi_c(\mathbf{x}_t, \mathbf{z}_t) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_t^T \mathbf{R}_t^{-1} \mathbf{H}_t & -\mathbf{H}_t^T \mathbf{R}_t^{-1} \\ -\mathbf{R}_t^{-1} \mathbf{H}_t & \mathbf{R}_t^{-1} \end{bmatrix} \right)$	
Iniciación de los potenciales de los conglomerados	
(1)	$\Phi_{\mathbf{x}_0 \mathbf{x}_1}^0(\mathbf{x}_0, \mathbf{x}_1) = \Phi_c(\mathbf{x}_0) \cdot \Phi_c(\mathbf{x}_0, \mathbf{x}_1) =$ $= \left(g, \begin{bmatrix} \hat{\mathbf{y}}_0 - \mathbf{F}_0^T (\mathbf{Q}_0(\mathbf{b}))^{-1} \mathbf{u}_0 \\ (\mathbf{Q}_0(\mathbf{b}))^{-1} \mathbf{u}_0 \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_0 + \mathbf{F}_0^T (\mathbf{Q}_0(\mathbf{b}))^{-1} \mathbf{F}_0 & -\mathbf{F}_0^T (\mathbf{Q}_0(\mathbf{b}))^{-1} \\ -(\mathbf{Q}_0(\mathbf{b}))^{-1} \mathbf{F}_0 & (\mathbf{Q}_0(\mathbf{b}))^{-1} \end{bmatrix} \right)$
(2)	$\Phi_{\mathbf{x}_t \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{x}_{t+1}) = \Phi_c(\mathbf{x}_t, \mathbf{x}_{t+1}) = \left(g, \begin{bmatrix} -\mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{b}))^{-1} \mathbf{u}_t \\ (\mathbf{Q}_t(\mathbf{b}))^{-1} \mathbf{u}_t \end{bmatrix}, \begin{bmatrix} \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{b}))^{-1} \mathbf{F}_t & -\mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{b}))^{-1} \\ -(\mathbf{Q}_t(\mathbf{b}))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(\mathbf{b}))^{-1} \end{bmatrix} \right)$ <p style="text-align: center;">con $t = 1 : n - 1$</p>
(3)	$\Phi_{\mathbf{x}_t \mathbf{z}_t}^0(\mathbf{x}_t, \mathbf{z}_t) = \Phi_c(\mathbf{x}_t, \mathbf{z}_t) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_t^T \mathbf{R}_t^{-1} \mathbf{H}_t & -\mathbf{H}_t^T \mathbf{R}_t^{-1} \\ -\mathbf{R}_t^{-1} \mathbf{H}_t & \mathbf{R}_t^{-1} \end{bmatrix} \right) \quad \text{con } t = 1 : n$
(4)	$\Phi_{\mathbf{x}_t \mathbf{z}_t \cap \mathbf{x}_{t-1} \mathbf{x}_t}^0(\mathbf{x}_t) = (g, [\mathbf{0}], [\mathbf{0}]) \quad \text{con } t = 1 : n$
(5)	$\Phi_{\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}}^0(\mathbf{x}_t) = (g, [\mathbf{0}], [\mathbf{0}]) \quad \text{con } t = 1 : n - 1$

Tabla C2-5. Árbol de unión asociado a la red bayesiana (b) y los valores de los potenciales de sus conglomerados (1,2 y 3) y separadores (4 y 5) tras el proceso de inicialización.

El potencial (1) de cada tabla se encuentra asociada al conglomerado inicial de transición de estados $\mathbf{x}_0 \mathbf{x}_1$, hay un potencial tipo (2) por cada uno de los conglomerados restantes de transición $\mathbf{x}_t \mathbf{x}_{t+1}$ y un potencial tipo (3) por cada uno de los conglomerados sensoriales $\mathbf{x}_t \mathbf{z}_t$ en todo instante de muestreo t . Además, y por cada separador $\mathbf{x}_t \mathbf{x}_{t+1} \cap \mathbf{x}_{t+1} \mathbf{z}_{t+1}$ entre un nodo de transición y otro de medida hay un potencial tipo (4), y por cada separador $\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}$ entre conglomerados de transición de estado hay un potencial tipo (5). En el caso de los conglomerados $\Phi_c^{etapa}(\mathbf{C}_{etapa})$ representa el valor del potencial asociado al conglomerado cuyo conjunto de variables iniciales es \mathbf{C} , para una de las etapas de la demostración, donde \mathbf{C}_{etapa} es el conjunto de variables de las que depende dicho potencial en esa etapa. En el caso de los separadores, $\Phi_{\mathbf{C}_i \cap \mathbf{C}_j}^{etapa}(\mathbf{S}_{etapa})$ representa el potencial asociado al separador que se encuentra en el enlace que une los conglomerados \mathbf{C}_i y \mathbf{C}_j .

para una de las *etapas* de la demostración, donde $\mathbf{S}_{\text{etapa}}$ es el conjunto de variables de las que depende dicho potencial en esa *etapa*.

La única diferencia existente entre los potenciales de cada red se encuentra en los potenciales asociados a las etapas de transición de estados, i.e. en los potenciales tipo (1) y (2), ya que en el primer caso (red bayesiana con nodos para la señal de control) la expresión del potencial $\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$ es función de la señal de control y en el segundo (red bayesiana donde se incluye la señal de control como el valor medio de la variable de estado tras la transición) el potencial $\Phi_{\mathbf{x}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{x}_{t+1})$ no depende de forma explícita de dicha variable aunque los campos de la tripleta asociada a este potencial sí dependan de forma implícita de la misma. La diferencia en el número de variables vectoriales de los que dependen los potenciales en cada caso hacen que la dimensión de los elementos vectoriales y matriciales de la tripleta del potencial de la red bayesiana (a) sea mayor, y por lo tanto, ocupen más espacio de memoria a la hora de utilizar el método de inferencia de Hugin.

Analizando el problema desde el punto de vista del control clásico, este inconveniente computacional no se encuentra justificado, ya que el objetivo no es estimar el valor de la señal de control \mathbf{u}_t a partir de la información sensorial, sino el estimar el estado de la variable \mathbf{x}_t a partir de la información sensorial y la señal de control. Por lo tanto, durante el proceso de inferencia los potenciales $\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$ del árbol de unión asociado a la red bayesiana (a) dejarán de depender de la variable \mathbf{u}_t , bien porque se conozca el valor de la señal de control de forma exacta y se introduzca en el algoritmo como evidencia, o bien porque aún desconociendo el valor de la señal de forma exacta sí se conozca su valor medio $\hat{\mathbf{u}}_t$ y covarianza \mathbf{U}_t , y se marginalice la variable de control de las expresiones potenciales en los que aparece.

Los dos procesos por los que desaparece la variable de control de los potenciales tipo (1) y tipo (2) de la tabla C2-4 se presentan a continuación, como primera etapa del algoritmo de inferencia, ya que el orden en el que se realizan las operaciones sobre los potenciales no cambia el resultado final del algoritmo. Para representar los potenciales tipo (1) y tipo (2) de la tabla C2-4 se puede utilizar la misma expresión, que se presenta a continuación:

$$\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) = \left(g, \begin{bmatrix} \mathbf{B}_t \\ \mathbf{U}_t^{-1} \hat{\mathbf{u}}_t \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{C}_t + \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}))^{-1} \mathbf{F}_t & \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}))^{-1} & -\mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}))^{-1} \\ (\mathbf{Q}_t(\mathbf{a}))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(\mathbf{a}))^{-1} + \mathbf{U}_t^{-1} & -(\mathbf{Q}_t(\mathbf{a}))^{-1} \\ -(\mathbf{Q}_t(\mathbf{a}))^{-1} \mathbf{F}_t & -(\mathbf{Q}_t(\mathbf{a}))^{-1} & (\mathbf{Q}_t(\mathbf{a}))^{-1} \end{bmatrix} \right) \quad (\text{C2.8})$$

$$\text{con } t = 0 : n - 1, \mathbf{B}_t = \begin{cases} \hat{\mathbf{y}}_0 & \text{si } t = 0 \\ \mathbf{0} & \text{si } t \neq 0 \end{cases} \text{ y } \mathbf{C}_t = \begin{cases} \mathbf{Y}_0 & \text{si } t = 0 \\ \mathbf{0} & \text{si } t \neq 0 \end{cases}$$

▪ **Incorporación de la señal de control como evidencia en el potencial (C2-8)**

En el caso de conocer el valor exacto de la señal de control, que denominaremos μ_t , la dependencia de los potenciales $\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$ de la variable aleatoria \mathbf{u}_t puede ser eliminada introduciendo dicho valor exacto como evidencia en el potencial. Las operaciones realizadas se presentan en las ecuaciones (C2-9), (C2-10), (C2-11) y (C2-12). La primera especifica las operaciones intermedias que hay que realizar, la segunda presenta la reorganización de las variables para poder introducir la evidencia en la última variable, en la tercera se introduce la evidencia y en la última se presenta el resultado final de todo el proceso. Si se compara el potencial final (C2-12), con las ecuaciones (1) y (2) de la tabla C2-5 se observa que cuando $\mathbf{Q}_t(a)=\mathbf{Q}_t(b)$ y $\mu_t=\mathbf{u}_t$ las expresiones son equivalentes, y por lo tanto también las dos redes lo son.

$$\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^1(\mathbf{x}_t, \mathbf{x}_{t+1}, \mu_t) = \Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) \Big|_{\mathbf{u}_t=\mu_t} = \Phi_{\mathbf{x}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{u}_t \Big|_{\mathbf{u}_t=\mu_t}) \quad (\text{C2.9})$$

$$\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{u}_t) = \left(g, \begin{bmatrix} \mathbf{B}_t \\ \mathbf{0} \\ \mathbf{U}_t^{-1} \hat{\mathbf{u}}_t \end{bmatrix}, \begin{bmatrix} \mathbf{C}_t + \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & -\mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} & \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \\ -(\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(a))^{-1} & -(\mathbf{Q}_t(a))^{-1} \\ (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & -(\mathbf{Q}_t(a))^{-1} & (\mathbf{Q}_t(a))^{-1} + \mathbf{U}_t^{-1} \end{bmatrix} \right) \quad (\text{C2.10})$$

$$\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{u}_t \Big|_{\mathbf{u}_t=\mu_t}) = \left(g, \begin{bmatrix} \mathbf{B}_t \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \\ -(\mathbf{Q}_t(a))^{-1} \end{bmatrix} \mu_t, \begin{bmatrix} \mathbf{C}_t + \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & -\mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \\ -(\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(a))^{-1} \end{bmatrix} \right) \quad (\text{C2.11})$$

$$\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^1(\mathbf{x}_t, \mathbf{x}_{t+1}, \mu_t) = \left(g, \begin{bmatrix} \mathbf{B}_t - \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \mu_t \\ (\mathbf{Q}_t(a))^{-1} \mu_t \end{bmatrix}, \begin{bmatrix} \mathbf{C}_t + \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & -\mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \\ -(\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(a))^{-1} \end{bmatrix} \right) \quad (\text{C2.12})$$

$$con \ t = 0 : n - 1, \mathbf{B}_t = \begin{cases} \hat{\mathbf{y}}_0 & si \ t = 0 \\ \mathbf{0} & si \ t \neq 0 \end{cases} \text{ y } \mathbf{C}_t = \begin{cases} \mathbf{Y}_0 & si \ t = 0 \\ \mathbf{0} & si \ t \neq 0 \end{cases}$$

▪ **Marginalización de la señal de control del potencial (C2-8)**

Para el caso en el que no se conoce el valor exacto de la señal de control si no sólo su valor medio $\hat{\mathbf{u}}_t$ y covarianza \mathbf{U}_t , la eliminación de la variable \mathbf{u}_t de los potenciales $\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1})$ se realiza marginalizando dicha variable de la expresión (C2-8). Las operaciones realizadas se presentan en las ecuaciones (C2-13), (C2-14), (C2-15) y (C2-16). La primera especifica las operaciones intermedias que hay que realizar, la segunda presenta la reorganización de las variables para poder marginalizar la primera de las mismas, en la tercera se marginaliza la variable y en la última se presenta el resultado final de todo el proceso.

$$\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^1(\mathbf{x}_t, \mathbf{x}_{t+1}) = \int \Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}) d\mathbf{u}_t = \int \Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{u}_t, \mathbf{x}_t, \mathbf{x}_{t+1}) d\mathbf{u}_t \quad (\text{C2.13})$$

$$\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}}^0(\mathbf{u}_t, \mathbf{x}_t, \mathbf{x}_{t+1}) = \left(g, \begin{bmatrix} \mathbf{U}_t^{-1} \hat{\mathbf{u}}_t \\ \mathbf{B}_t \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} (\mathbf{Q}_t(a))^{-1} + \mathbf{U}_t^{-1} & (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & -(\mathbf{Q}_t(a))^{-1} \\ \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} & \mathbf{C}_t + \mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & -\mathbf{F}_t^T (\mathbf{Q}_t(a))^{-1} \\ -(\mathbf{Q}_t(a))^{-1} & -(\mathbf{Q}_t(a))^{-1} \mathbf{F}_t & (\mathbf{Q}_t(a))^{-1} \end{bmatrix} \right) \quad (\text{C2.14})$$

$$\int \Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+I}}^0(\mathbf{u}_t, \mathbf{x}_t, \mathbf{x}_{t+I}) d\mathbf{u}_t = (\mathbf{g}, \mathbf{h}^*, \mathbf{K}^*) = \left(\mathbf{g}, \mathbf{h}^*, \begin{bmatrix} \mathbf{K}_{11}^* & \mathbf{K}_{12}^* \\ \mathbf{K}_{21}^* & \mathbf{K}_{22}^* \end{bmatrix} \right) \quad (\text{C2.15})$$

$$\begin{aligned} \mathbf{h}^* &= \begin{bmatrix} \mathbf{B}_t \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}))^{-1} \\ -(\mathbf{Q}_t(\mathbf{a}))^{-1} \end{bmatrix} \left((\mathbf{Q}_t(\mathbf{a}))^{-1} + \mathbf{U}_t^{-1} \right)^{-1} \mathbf{U}_t^{-1} \hat{\mathbf{u}}_t = \\ &= \begin{bmatrix} \mathbf{B}_t \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_t^T \\ -1 \end{bmatrix} (\mathbf{Q}_t(\mathbf{a}))^{-1} \left(\mathbf{U}_t - \mathbf{U}_t (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \mathbf{U}_t \right) \mathbf{U}_t^{-1} \hat{\mathbf{u}}_t = \\ &= \begin{bmatrix} \mathbf{B}_t \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_t^T \\ -1 \end{bmatrix} (\mathbf{Q}_t(\mathbf{a}))^{-1} \left(1 - \mathbf{U}_t (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \right) \hat{\mathbf{u}}_t = \\ &= \begin{bmatrix} \mathbf{B}_t \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_t^T \\ -1 \end{bmatrix} (\mathbf{Q}_t(\mathbf{a}))^{-1} \left((\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)(\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} - \mathbf{U}_t (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \right) \hat{\mathbf{u}}_t = \\ &= \begin{bmatrix} \mathbf{B}_t \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_t^T \\ -1 \end{bmatrix} (\mathbf{Q}_t(\mathbf{a}))^{-1} (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t - \mathbf{U}_t)(\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \hat{\mathbf{u}}_t = \\ &= \begin{bmatrix} \mathbf{B}_t \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{F}_t^T \\ -1 \end{bmatrix} (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \hat{\mathbf{u}}_t = \begin{bmatrix} \mathbf{B}_t - \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \hat{\mathbf{u}}_t \\ (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \hat{\mathbf{u}}_t \end{bmatrix} \end{aligned}$$

$$\mathbf{K}_{11}^* = \mathbf{C}_t + \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}))^{-1} \mathbf{F}_t - \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}))^{-1} \left((\mathbf{Q}_t(\mathbf{a}))^{-1} + \mathbf{U}_t^{-1} \right)^{-1} (\mathbf{Q}_t(\mathbf{a}))^{-1} \mathbf{F}_t = \mathbf{C}_t + \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \mathbf{F}_t$$

$$\mathbf{K}_{12}^* = -\mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}))^{-1} \mathbf{F}_t + \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}))^{-1} \left((\mathbf{Q}_t(\mathbf{a}))^{-1} + \mathbf{U}_t^{-1} \right)^{-1} (\mathbf{Q}_t(\mathbf{a}))^{-1} = -\mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1}$$

$$\mathbf{K}_{21}^* = -(\mathbf{Q}_t(\mathbf{a}))^{-1} \mathbf{F}_t + (\mathbf{Q}_t(\mathbf{a}))^{-1} \left((\mathbf{Q}_t(\mathbf{a}))^{-1} + \mathbf{U}_t^{-1} \right)^{-1} (\mathbf{Q}_t(\mathbf{a}))^{-1} \mathbf{F}_t = -(\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \mathbf{F}_t$$

$$\mathbf{K}_{22}^* = (\mathbf{Q}_t(\mathbf{a}))^{-1} - (\mathbf{Q}_t(\mathbf{a}))^{-1} \left((\mathbf{Q}_t(\mathbf{a}))^{-1} + \mathbf{U}_t^{-1} \right)^{-1} (\mathbf{Q}_t(\mathbf{a}))^{-1} = (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1}$$

$$\Phi_{\mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+I}}^1(\mathbf{x}_t, \mathbf{x}_{t+I}) = \quad (\text{C2.16})$$

$$= \left(\mathbf{g}, \begin{bmatrix} \mathbf{B}_t - \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \mathbf{u}_t \\ (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \mathbf{u}_t \end{bmatrix}, \begin{bmatrix} \mathbf{C}_t + \mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \mathbf{F}_t & -\mathbf{F}_t^T (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \\ -(\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \mathbf{F}_t & (\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t)^{-1} \end{bmatrix} \right)$$

$$\text{con } t = 0 : n - I, \mathbf{B}_t = \begin{cases} \hat{\mathbf{y}}_0 & \text{si } t = 0 \\ \mathbf{0} & \text{si } t \neq 0 \end{cases} \text{ y } \mathbf{C}_t = \begin{cases} \mathbf{Y}_0 & \text{si } t = 0 \\ \mathbf{0} & \text{si } t \neq 0 \end{cases}$$

Si se compara el potencial final (C2-16) con el de las ecuaciones (1) y (2) de la tabla C2-5 se observa, como cabría esperar tras un análisis directo de las expresiones de probabilidad condicional en forma normal, que las ecuaciones son iguales cuando $\mathbf{Q}_t(\mathbf{a}) + \mathbf{U}_t = \mathbf{Q}_t(\mathbf{b})$ y $\hat{\mathbf{u}}_t = \mathbf{u}_t$. Por lo tanto, cuando no se sabe el valor exacto de la señal de control se puede utilizar la red bayesiana (b) igualando la covarianza de la ecuación condicional de transición de estados a la suma de las covarianzas debidas a la transición de estados y a la señal de control.

Por lo tanto, la forma de modelar la existencia de la señal de control en ambas redes es equivalente a la hora de realizar inferencia sobre el estado de la variable del sistema \mathbf{x}_t . A pesar de dicha equivalencia, la segunda red es estructuralmente más sencilla y el proceso de inferencia necesita realizar menos cálculos, *por lo que la dependencia que existe entre el estado y la señal de control se modelará como en la red bayesiana (b) en el resto de las demostraciones de este capítulo.*

C2.2.2. Inferencia exacta y fusión sensorial asíncrona y/o temporalmente retardada sobre la red bayesiana que modela un sistema lineal de control con múltiples sensores

En este apartado se estudia el comportamiento del proceso de inferencia exacta desde el punto de vista sensorial sobre la red bayesiana que modela un sistema lineal de control cuyo estado \mathbf{x}_t es observado por múltiples sensores. El funcionamiento de estos sensores puede ser síncrono (caso en el que todos los sensores miden a la vez), asíncrono (caso en el que cada sensor mide a distintas velocidades), y en ambos casos los sensores pueden proporcionar la medida con un retardo temporal variable, debido a retardos en el procesamiento o en el proceso de comunicación de la medida. Los objetivos de este estudio son: (1) comprobar que el comportamiento del algoritmo de inferencia exacta sobre la red propuesta para modelar el sistema es equivalente al comportamiento del filtro de Información distribuido, (2) analizar la forma en la que el algoritmo almacena la información sensorial, (3) observar como se incorpora en los potenciales la información que llega a la red temporalmente retardada, y (4) destacar cuales son las operaciones de absorción mínimas necesarias para que el comportamiento del algoritmo de inferencia sea equivalente al del filtro de Información. Para este análisis modelaremos la existencia de la señal de control igual que en la red bayesiana (b) del apartado anterior, ya que esa forma hace que el algoritmo de inferencia necesite realizar menos cálculos.

La red bayesiana propuesta en este apartado modela el comportamiento de un sistema de control lineal discreto cuyo estado \mathbf{x}_{t+1} en el instante $t+1$: (1) depende del estado \mathbf{x}_t en el instante t y de la señal de control aplicada \mathbf{u}_t durante el periodo de muestreo, y (2) es observado por un conjunto de S sensores, que proporcionan en ese instante las medidas $\mathbf{z}_{s,t+1}$. Los sensores pueden funcionar de forma síncrona (cuando todos los sensores proporcionan una medida en todos los instantes de muestreo), asíncrona (caso en el que no habrá una medida disponible en todos los instantes de muestreo para todos los sensores) y proporcionar las medidas al algoritmo de inferencia con un retardo variable. Los modelos del sistema (C2-17) y sensoriales (C2-18) son funciones lineales no deterministas, que incluyen un término aditivo responsable de modelar ruido blanco (\mathbf{v}_t y $\mathbf{v}_{s,t+1}$) cuyas covarianzas son \mathbf{Q}_t y $\mathbf{R}_{s,t+1}$:

$$\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{u}_t + \mathbf{v}_t \quad (\text{C2.17})$$

$$\mathbf{z}_{s,t+1} = \mathbf{H}_{s,t+1} \mathbf{x}_{s,t+1} + \mathbf{v}_{s,t+1} \quad (\text{C2.18})$$

Al igual que en el apartado anterior, el objetivo del problema es estimar el estado del sistema y su covarianza a partir de un estado y covarianza iniciales, las ordenes de control utilizadas y la información sensorial disponible.

La red bayesiana recogida en la tabla C2-6 permite modelar y resolver este problema para el caso en el que hay tres sensores que proporcionan medidas $\mathbf{z}_{s,t}$ independientes entre sí y relacionadas linealmente con el estado del sistema \mathbf{x}_t . En el caso de tener un número diferente de sensores sólo es necesario igualar a dicho número el número de nodos sensoriales de la red para cada instante de muestreo. La estructura de esta red, aunque soporta la existencia de medidas de todos los sensores para cada instante de muestreo, no implica que este hecho tenga que ocurrir necesariamente. Los sensores pueden proporcionar medidas de forma asíncrona, y en el caso en el que un sensor no proporcione una medida para un instante determinado (por ejemplo, el sensor 2 en el instante 1), bastará con no introducir evidencia en los conglomerados sensoriales del árbol de unión que incluyan la variable asociada a esa medida (para el ejemplo, el conglomerado con la variable $\mathbf{z}_{2,1}$).

Planteamiento del Problema	
$\Pr(\mathbf{x}_0) = N_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, \mathbf{P}_0) \quad \Pr(\mathbf{x}_{t+1} \mathbf{x}_t) = N_{\mathbf{x}_{t+1}}(\mathbf{u}_t + \mathbf{F}_t \mathbf{x}_t, \mathbf{Q}_t) \quad \Pr(\mathbf{z}_{s,t} \mathbf{x}_t) = N_{\mathbf{z}_{s,t}}(\mathbf{0} + \mathbf{H}_{s,t} \mathbf{x}_t, \mathbf{R}_{s,t})$	
<p>El objetivo es calcular</p> $\hat{\mathbf{x}}_n \setminus \{\hat{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{z}_{s,t} = \zeta_{s,t}, \mathbf{H}_{s,t}, \mathbf{R}_{s,t}, \mathbf{u}_{t-1}, \mathbf{F}_{t-1} ; s=1:S, t=1:n\} \quad \text{y} \quad \mathbf{P}_n \setminus \{\hat{\mathbf{x}}_0, \mathbf{P}_0, \mathbf{z}_{s,t} = \zeta_{s,t}, \mathbf{H}_{s,t}, \mathbf{R}_{s,t}, \mathbf{u}_{t-1}, \mathbf{F}_{t-1} ; s=1:S, t=1:n\}$	

Tabla C2-6. Red bayesiana para modelar un sistema lineal de control sobre el que $S=3$ sensores pueden realizar medidas de forma acíclica y temporalmente retardada.

Antes de comenzar el proceso de inferencia exacta sobre esta red bayesiana (en el que durante las demostraciones no se calculará la g de los potenciales canónicos) es necesario:

- Construir su árbol de unión, labor que no encierra ninguna dificultad en este caso, ya que en la etapa de moralización se obtiene directamente un grafo ya triangulado muy sencillo, en el que es directo identificar los nodos asociados a cada conglomerado y obtener un árbol de conglomerados que cumpla la propiedad de intersección dinámica. El árbol de unión asociado a la red se puede ver en la parte superior de la tabla C2-7, utilizando para representar a los conglomerados y separadores nodos elípticos y rectangulares respectivamente.
- Convertir las expresiones de probabilidad condicionales que modelan las relaciones entre las diferentes variables de la red en potenciales condicionales canónicos Φ_c , de acuerdo con la expresión de inicialización de la tabla C2-2. El resultado de este proceso se encuentra recogido en la parte intermedia de la tabla C2-7. El elemento vectorial y matricial del potencial asociado al estado inicial $\Phi_c(\mathbf{x}_0)$ almacena, de acuerdo con la nomenclatura seguida en el filtro de Información, la información del estado inicial $\hat{\mathbf{y}}_0 = \mathbf{Y}_0 \hat{\mathbf{x}}_0$ y la

covarianza asociada a dicha información $\mathbf{Y}_0 = \mathbf{P}_0^{-1}$. En el apéndice I se analiza la relación existente entre el espacio de estados (determinado por las variables $\hat{\mathbf{x}}_0$ y \mathbf{P}_0) y el espacio de información sobre el estado (determinado por las variables $\hat{\mathbf{y}}_0$ e \mathbf{Y}_0).

- Inicializar las expresiones de los potenciales de los conglomerados (como el producto de los potenciales condicionales canónicos Φ_c en los que aparecen subconjuntos de los nodos del conglomerado, reorganizados y ampliados en el caso que sea necesario, de acuerdo con las operaciones recogidas en la tabla C2-2) y de los separadores (asignándole un 0, escalar, vectorial y matricial respectivamente, a los tres campos de la tripleta del potencial). El resultado de este proceso se encuentra en la parte inferior de la tabla C2-7.

Árbol de conglomerados del problema	
Conversión de los potenciales normales en canónicos	
$\Pr(\mathbf{x}_0) = N_{\mathbf{x}_0}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$	$\Rightarrow \Phi_c(\mathbf{x}_0) = (g, \mathbf{P}_0^{-1}\hat{\mathbf{x}}_0, \mathbf{P}_0^{-1}) = (g, \hat{\mathbf{y}}_0, \mathbf{Y}_0)$
$\Pr(\mathbf{x}_{t+1} \mathbf{x}_t) = N_{\mathbf{x}_{t+1}}(\mathbf{u}_t + \mathbf{F}_t \mathbf{x}_t, \mathbf{Q}_t)$	$\Rightarrow \Phi_c(\mathbf{x}_t, \mathbf{x}_{t+1}) = \left(g, \begin{bmatrix} -\mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{u}_t \\ \mathbf{Q}_t^{-1} \mathbf{u}_t \end{bmatrix}, \begin{bmatrix} \mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{F}_t & -\mathbf{F}_t^T \mathbf{Q}_t^{-1} \\ -\mathbf{Q}_t^{-1} \mathbf{F}_t & \mathbf{Q}_t^{-1} \end{bmatrix} \right)$
$\Pr(\mathbf{z}_{s,t} \mathbf{x}_t) = N_{\mathbf{z}_{s,t}}(\mathbf{0} + \mathbf{H}_{s,t} \mathbf{x}_t, \mathbf{R}_{s,t})$	$\Rightarrow \Phi_c(\mathbf{x}_t, \mathbf{z}_{s,t}) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & -\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right)$
Inicialización de los potenciales de los conglomerados	
(1)	$\Phi_{\mathbf{x}_0 \mathbf{x}_1}^0(\mathbf{x}_0, \mathbf{x}_1) = \Phi_c(\mathbf{x}_0) \cdot \Phi_c(\mathbf{x}_0, \mathbf{x}_1) = \left(g, \begin{bmatrix} \hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \\ \mathbf{Q}_0^{-1} \mathbf{u}_0 \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 & -\mathbf{F}_0^T \mathbf{Q}_0^{-1} \\ -\mathbf{Q}_0^{-1} \mathbf{F}_0 & \mathbf{Q}_0^{-1} \end{bmatrix} \right)$
(2)	$\Phi_{\mathbf{x}_t \mathbf{x}_{t+1}}^0(\mathbf{x}_t, \mathbf{x}_{t+1}) = \Phi_c(\mathbf{x}_t, \mathbf{x}_{t+1}) = \left(g, \begin{bmatrix} -\mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{u}_t \\ \mathbf{Q}_t^{-1} \mathbf{u}_t \end{bmatrix}, \begin{bmatrix} \mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{F}_t & -\mathbf{F}_t^T \mathbf{Q}_t^{-1} \\ -\mathbf{Q}_t^{-1} \mathbf{F}_t & \mathbf{Q}_t^{-1} \end{bmatrix} \right) \quad \text{con } t = 1 : n - 1$
(3)	$\Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^0(\mathbf{x}_t, \mathbf{z}_{s,t}) = \Phi_c(\mathbf{x}_t, \mathbf{z}_{s,t}) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & -\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right) \quad \text{con } t = 1 : n, s = 1 : S$
(4)	$\Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1} \mathbf{x}_t}^0(\mathbf{x}_t) = (0, [\mathbf{0}], [\mathbf{0}]) \quad \text{con } t = 1 : n, s = 1 : S$
(5)	$\Phi_{\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}}^0(\mathbf{x}_t) = (0, [\mathbf{0}], [\mathbf{0}]) \quad \text{con } t = 1 : n - 1$

Tabla C2-7. Árbol de unión de la red de la tabla C2-6, para un sistema con S=3 sensores, y sus potenciales asociados, tanto para los diferentes conglomerados (1, 2 y 3) como para los separadores (4 y 5), después de su inicialización.

La nomenclatura utilizada para los potenciales de los conglomerados y separadores durante las demostraciones es la siguiente. Para los conglomerados $\Phi_{\mathbf{C}}^{\text{etapa}}(\mathbf{C}_{\text{etapa}})$ representa el potencial asociado al conglomerado cuyo conjunto de variables iniciales es \mathbf{C} , para una de las etapas de la demostración, donde $\mathbf{C}_{\text{etapa}}$ es el conjunto de variables de las que depende dicho potencial en esa etapa. En el caso de los separadores $\Phi_{\mathbf{C}_i \cap \mathbf{C}_j}^{\text{etapa}}(\mathbf{S}_{\text{etapa}})$ representa el potencial asociado al separador

que se encuentra en el enlace que une los conglomerados C_i y C_j , para una de las *etapas* de la demostración, donde S_{etapa} es el conjunto de variables de las que depende dicho potencial en esa *etapa*. De esta forma, el potencial (1) corresponde al conglomerado asociado al conglomerado inicial de transición de estados x_0x_1 , los potenciales tipo (2) a los conglomerados asociados al resto de las transiciones de estado $x_t x_{t+1}$, los tipo (3) a los conglomerados sensoriales asociados a cada sensor en cada instante $x_t z_{s,t}$, los tipo (4) a los separadores entre cada conglomerado transición y de medida $x_{t-1}x_t \cap x_t z_{s,t}$, y los tipo (5) a los separadores entre conglomerados de transición de estado $x_{t-1}x_t \cap x_t x_{t+1}$.

Una vez realizada la inicialización comenzaremos con el proceso de inferencia. Como el filtro de Información distribuido utiliza para estimar el estado en un instante de tiempo solamente la información sensorial de ese instante y el estado del instante anterior (que a su vez almacena toda la información sensorial previa), si se desea obtener un algoritmo equivalente de propagación exacto sobre la red, las operaciones de absorción que se realicen sobre el árbol de unión no pueden propagar la evidencia de forma que el estado para un instante de muestreo utilice información sensorial de instantes de muestro posteriores. Este último tipo de absorciones se encuentran fuera del ámbito de cálculo del filtro de Información, estando relacionadas directamente con las operaciones de alisado, proceso que sobrecargaría computacionalmente el algoritmo de inferencia sin permitirnos mejorar la estima del estado en el último instante de muestreo, que es habitualmente el estado de interés en los sistemas de control en tiempo real.

Por lo tanto, para obtener un algoritmo de inferencia equivalente al del filtro de Información, las únicas operaciones de absorción que se pueden realizar entre los nodos de transición adyacentes del árbol de unión son aquellas en las que un nodo de la derecha absorba información de uno de la izquierda. Es decir entre los nodos transición de estados se pueden realizar las absorciones tipo (1) y no tipo (2) representadas en el esquema de la figura C2-4. Sin embargo, aunque las operaciones de absorción entre los nodos de transición y los nodos sensoriales se pueden realizar en los dos sentidos (absorciones tipo (3) y tipo (4) de la figura C2-4), cuando no se desea estimar el valor de la medida de un sensor antes de que dicha medida se produzca, las absorciones tipo (4) no son realmente necesarias y únicamente aumentan el número de operaciones del algoritmo. Por lo tanto, el algoritmo de inferencia presentado en esta sección sólo mandará mensajes de absorción de izquierda a derecha y de abajo a arriba.

Para ver como el algoritmo incorpora y almacena la evidencia proporcionada de forma retardada y temporalmente desordenada por múltiples sensores, y como se estima el estado del sistema x_t a partir de dicha evidencia se usará un caso hipotético, en el que se supone:

- que ninguno de los S sensores es capaz de proporcionar medidas antes del instante de muestreo n , por lo que la mejor estima del estado para todos los instantes antes de que haya ninguna medida disponible es la que se obtiene propagando hacia delante únicamente la información asociada al estado inicial y a las señales de control.
- que una vez que se ha obtenido la estima del estado en el instante n sin ninguna información sensorial llega la información de todos los sensores y es introducida en los nodos sensoriales del árbol de unión y propagada de forma que ésta sea utilizada para mejorar la estima del valor de cada estado con toda la información sensorial que no sea posterior a su instante de muestreo.

De esta forma, como la información sensorial se introduce una vez que ya se ha estimado el estado del sistema sin la misma, se puede observar al mismo tiempo como la información sensorial modifica los potenciales del árbol de unión en el caso distribuido y como se corrige la estima del estado del sistema con información sensorial proporcionada al algoritmo de inferencia temporalmente retardada. El tratamiento de información sensorial que llega al sistema de forma temporalmente desordenada no está directamente recogido en el caso hipotético propuesto, aunque en la última sección de este apartado se explicará como una generalización del mismo justifica la equivalencia del algoritmo de propagación exacto sobre el árbol de unión y el filtro de Información distribuido.

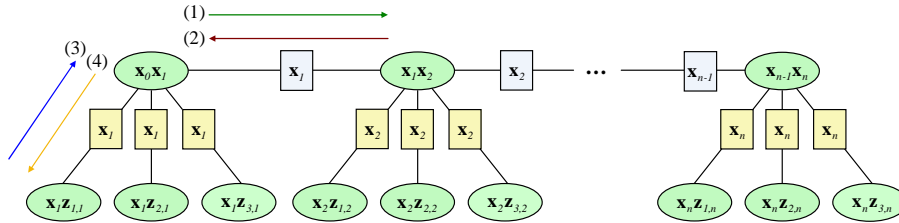


Figura C2-4. Árbol de unión y tipos de absorciones del algoritmo de inferencia para conseguir la equivalencia de dicho algoritmo y el del filtro de Información distribuido: los tipos (1) y (3) son necesarios, el tipo (4) es optativo, y el tipo (2) está prohibido.

En las secciones siguientes se presentan las operaciones realizadas por el algoritmo de inferencia exacta sobre los potenciales para este caso hipotético y la generalización del ejemplo para el caso en el que la información llega de forma desordenada al algoritmo de inferencia.

▪ **Estimación del estado del sistema para todos los instantes mientras no existe ninguna evidencia sensorial**

En este caso, el mínimo número de operaciones de absorción necesarias para el algoritmo de inferencia se consigue cuando se propaga a través de los nodos de transición, hacia delante y de forma ordenada con mensajes probabilísticos del tipo (1) de la figura C2-4, la información sobre el

estado inicial y sobre la señal de control. No es necesario realizar de momento absorciones del tipo (3), ya que al no existir ninguna evidencia dichas operaciones no modificarán el estado de los potenciales de los conglomerados de transición de estados.

La primera absorción se da entre los conglomerados de transición $\mathbf{x}_0\mathbf{x}_1$ y $\mathbf{x}_1\mathbf{x}_2$ y sus operaciones son recogidas en las siguientes expresiones: (C2-19) marginaliza el potencial asociado al conglomerado $\mathbf{x}_0\mathbf{x}_1$ sobre la variable \mathbf{x}_0 para obtener el valor del potencial del separador $\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{x}_2$ que es actualizado en (C2-20), y (C2-21) modifica el potencial del conglomerado $\mathbf{x}_1\mathbf{x}_2$ con la información procedente del separador.

$$\Phi_{\mathbf{x}_0\mathbf{x}_1}^{0*}(\mathbf{x}_1) = \int \Phi_{\mathbf{x}_0\mathbf{x}_1}^0(\mathbf{x}_0, \mathbf{x}_1) d\mathbf{x}_0 = (\mathbf{g}, \mathbf{h}^*, \mathbf{K}^*) = (\mathbf{g}, [\hat{\mathbf{y}}_1(1)], [\mathbf{Y}_1(1)]) \quad (\text{C2.19})$$

$$\begin{aligned} \mathbf{K}^* &= \mathbf{Q}_0^{-1} - \mathbf{Q}_0^{-1}\mathbf{F}_0 \left(\mathbf{Y}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 \right)^{-1} \mathbf{F}_0^T \mathbf{Q}_0^{-1} = \left(\mathbf{F}_0 \mathbf{Y}_0^{-1} \mathbf{F}_0^T + \mathbf{Q}_0 \right)^{-1} = \mathbf{Y}_1(1) \\ \mathbf{h}^* &= \mathbf{Q}_0^{-1} \mathbf{u}_0 + \mathbf{Q}_0^{-1} \mathbf{F}_0 \left(\mathbf{Y}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 \right)^{-1} \left(\hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \right) = \\ &= \mathbf{Q}_0^{-1} \mathbf{u}_0 + \mathbf{Q}_0^{-1} \mathbf{F}_0 \left(\mathbf{Y}_0^{-1} - \mathbf{Y}_0^{-1} \mathbf{F}_0^T \left(\mathbf{F}_0 \mathbf{Y}_0^{-1} \mathbf{F}_0^T + \mathbf{Q}_0 \right)^{-1} \mathbf{F}_0 \mathbf{Y}_0^{-1} \right) \left(\hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \right) = \\ &= \mathbf{Q}_0^{-1} \mathbf{u}_0 + \mathbf{Q}_0^{-1} \left(\mathbf{F}_0 \mathbf{Y}_0^{-1} - \mathbf{F}_0 \mathbf{Y}_0^{-1} \mathbf{F}_0^T \mathbf{Y}_1(1) \mathbf{F}_0 \mathbf{Y}_0^{-1} \right) \left(\hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \right) = \\ &= \mathbf{Q}_0^{-1} \mathbf{u}_0 + \mathbf{Q}_0^{-1} \left(\mathbf{1} - \left(\mathbf{Y}_1(1) \right)^{-1} - \mathbf{Q}_0 \right) \mathbf{Y}_1(1) \mathbf{F}_0 \mathbf{Y}_0^{-1} \left(\hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \right) = \\ &= \mathbf{Q}_0^{-1} \mathbf{u}_0 + \mathbf{Y}_1(1) \mathbf{F}_0 \mathbf{Y}_0^{-1} \left(\hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \right) = \mathbf{Y}_1(1) \mathbf{F}_0 \mathbf{Y}_0^{-1} \hat{\mathbf{y}}_0 + \left(\mathbf{1} - \mathbf{Y}_1(1) \mathbf{F}_0 \mathbf{Y}_0^{-1} \mathbf{F}_0^T \right) \mathbf{Q}_0^{-1} \mathbf{u}_0 = \\ &= \hat{\mathbf{y}}_1(1) \mathbf{F}_0 \mathbf{Y}_0^{-1} \hat{\mathbf{y}}_0 + \left(\mathbf{1} - \mathbf{Y}_1(1) \left(\left(\mathbf{Y}_1(1) \right)^{-1} - \mathbf{Q}_0 \right) \right) \mathbf{Q}_0^{-1} \mathbf{u}_0 = \mathbf{Y}_1(1) \mathbf{F}_0 \mathbf{Y}_0^{-1} \hat{\mathbf{y}}_0 + \mathbf{Y}_1(1) \mathbf{u}_0 = \\ &= \mathbf{Y}_1(1) \left(\mathbf{F}_0 \mathbf{Y}_0^{-1} \hat{\mathbf{y}}_0 + \mathbf{u}_0 \right) = \hat{\mathbf{y}}_1(1) \end{aligned}$$

$$\begin{aligned} \Phi_{\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{x}_2}^1(\mathbf{x}_1) &= \Phi_{\mathbf{x}_0\mathbf{x}_1}^{0*}(\mathbf{x}_1) = (\mathbf{g}, [\hat{\mathbf{y}}_1(1)], [\mathbf{Y}_1(1)]) = \\ &= \left(\mathbf{g}, \left[\left(\mathbf{F}_0 \mathbf{Y}_0^{-1} \mathbf{F}_0^T + \mathbf{Q}_0 \right)^{-1} \left(\mathbf{F}_0 \mathbf{Y}_0^{-1} \hat{\mathbf{y}}_0 + \mathbf{u}_0 \right) \right], \left[\left(\mathbf{F}_0 \mathbf{Y}_0^{-1} \mathbf{F}_0^T + \mathbf{Q}_0 \right)^{-1} \right] \right) \end{aligned} \quad (\text{C2.20})$$

$$\begin{aligned} \Phi_{\mathbf{x}_1\mathbf{x}_2}^1(\mathbf{x}_1, \mathbf{x}_2) &= \Phi_{\mathbf{x}_1\mathbf{x}_2}^0(\mathbf{x}_1, \mathbf{x}_2) \cdot \frac{\Phi_{\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{x}_2}^1(\mathbf{x}_1)}{\Phi_{\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{x}_2}^0(\mathbf{x}_1)} = \\ &= \left(\mathbf{g}, \begin{bmatrix} -\mathbf{F}_1^T \mathbf{Q}_1^{-1} \mathbf{u}_1 \\ \mathbf{Q}_1^{-1} \mathbf{u}_1 \end{bmatrix}, \begin{bmatrix} \mathbf{F}_1^T \mathbf{Q}_1^{-1} \mathbf{F}_1 & -\mathbf{F}_1^T \mathbf{Q}_1^{-1} \\ -\mathbf{Q}_1^{-1} \mathbf{F}_1 & \mathbf{Q}_1^{-1} \end{bmatrix} \right) \cdot \frac{(\mathbf{g}, [\hat{\mathbf{y}}_1(1)], [\mathbf{Y}_1(1)])}{(\emptyset, [\mathbf{0}], [\mathbf{0}])} = \\ &= \left(\mathbf{g}, \begin{bmatrix} \hat{\mathbf{y}}_1(1) - \mathbf{F}_1^T \mathbf{Q}_1^{-1} \mathbf{u}_1 \\ \mathbf{Q}_1^{-1} \mathbf{u}_1 \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_1(1) + \mathbf{F}_1^T \mathbf{Q}_1^{-1} \mathbf{F}_1 & -\mathbf{F}_1^T \mathbf{Q}_1^{-1} \\ -\mathbf{Q}_1^{-1} \mathbf{F}_1 & \mathbf{Q}_1^{-1} \end{bmatrix} \right) \end{aligned} \quad (\text{C2.21})$$

El nuevo valor del potencial del separador, obtenido en la marginalización (C2-19), almacena el valor estimado de la información del estado después de la etapa de predicción del filtro de Información $\hat{\mathbf{y}}_1(1)$, y la covarianza asociada a la misma $\mathbf{Y}_1(1)$.

Los resultados del resto de las absorciones se desprenden directamente de la similitud existente entre el resultado de la fórmula (C2-21), donde se presenta el potencial que habría que marginalizar en la siguiente absorción del algoritmo, y la expresión (1) de la tabla C2-7, que recoge el potencial que se ha marginalizado en la primera absorción (C2-19). Esta semejanza hace que se pueden

realizar sucesivamente operaciones similares a las de las fórmulas (C2-19), (C2-20) y (C2-21), obteniéndose para cada conglomerado y separador los potenciales recogidos en la tabla siguiente.

Árbol de conglomerados del problema	
Potenciales tras propagar únicamente hacia delante información sobre el estado inicial y sobre las señales de control	
(1)	$\Phi_{x_0 x_1}^1(x_0, x_1) = \Phi_{x_0 x_1}^0(x_0, x_1) = \left(g, \begin{bmatrix} \hat{y}_0 - F_0^T Q_0^{-1} u_0 \\ Q_0^{-1} u_0 \end{bmatrix}, \begin{bmatrix} Y_0 + F_0^T Q_0^{-1} F_0 & -F_0^T Q_0^{-1} \\ -Q_0^{-1} F_0 & Q_0^{-1} \end{bmatrix} \right)$
(2)	$\Phi_{x_t x_{t+1}}^1(x_t, x_{t+1}) = \Phi_{x_t x_{t+1}}^0(x_t, x_{t+1}) \cdot \frac{\Phi_{x_{t-1} x_t \cap x_t x_{t+1}}^1(x_t)}{\Phi_{x_{t-1} x_t \cap x_t x_{t+1}}^0(x_t)} =$ $= \left(g, \begin{bmatrix} \hat{y}_t(1) - F_t^T Q_t^{-1} u_t \\ Q_t^{-1} u_t \end{bmatrix}, \begin{bmatrix} Y_t(1) + F_t^T Q_t^{-1} F_t & -F_t^T Q_t^{-1} \\ -Q_t^{-1} F_t & Q_t^{-1} \end{bmatrix} \right) \quad \text{con } t = 1 : n - 1$
(3)	$\Phi_{x_t z_{s,t}}^1(x_t, z_{s,t}) = \Phi_{x_t z_{s,t}}^0(x_t, z_{s,t}) = \left(g, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} H_{s,t}^T R_{s,t}^{-1} H_{s,t} & -H_{s,t}^T R_{s,t}^{-1} \\ -R_{s,t}^{-1} H_{s,t} & R_{s,t}^{-1} \end{bmatrix} \right) \quad \text{con } t = 1 : n, s = 1 : S$
(4)	$\Phi_{x_t z_{s,t} \cap x_{t-1} x_t}^1(x_t) = \Phi_{x_t z_{s,t} \cap x_{t-1} x_t}^0(x_t) = (0, [0], [0]) \quad \text{con } t = 1 : n, s = 1 : S$
(5)	$\Phi_{x_{t-1} x_t \cap x_t x_{t+1}}^1(x_t) = \Phi_{x_{t-1} x_t}^{0*}(x_t) = (g, [\hat{y}_t(1)], [Y_t(1)]) =$ $= \left(g, \left[(F_{t-1} Y_{t-1}^{-1}(1) F_{t-1}^T + Q_{t-1})^{-1} (F_{t-1} Y_{t-1}^{-1}(1) \hat{y}_{t-1}(1) + u_{t-1}) \right], \left[(F_{t-1} Y_{t-1}^{-1}(1) F_{t-1}^T + Q_{t-1})^{-1} \right] \right)$ <p style="text-align: center;">con $t = 1 : n - 1$, $\hat{y}_0(1) = \hat{y}_0$ e $Y_0(1) = Y_0$</p>

Tabla C2-8. Árbol de unión de la red de la tabla C2-6, con S=3 sensores, y sus potenciales asociados, tanto para los diferentes conglomerados (1, 2 y 3) como para los separadores (4 y 5), después de propagar hacia delante la información existente en el árbol de unión sobre el estado inicial y las señales de control.

Tras la primera etapa completa de inferencia, se puede observar que los potenciales de los conglomerados de las transiciones ((1) y tipo (2) de la tabla C2-8) pasan a tener una forma semejante; que los potenciales de los separadores que hay entre los conglomerados de transición (tipo (5)) almacenan, para cada instante de muestreo, la predicción sucesiva del valor estimado de la información del estado $\hat{y}_t(1)$ y su covarianza $Y_t(1)$; y que el resto de los potenciales (tipo (3) y tipo (4)) mantiene el valor asignado durante la etapa inicial, ya que hemos decidido no realizar de momento absorciones tipo (3) de la figura C2-4, ya que éstas no modificarían la estima de la información del estado (ni su covarianza).

Por razones de notación, el valor estimado de la información del estado $\hat{y}_t(1)$ y su covarianza $Y_t(1)$ se representa junto a un valor entre paréntesis que sirve para identificar la etapa completa de inferencia en la que han sido obtenidos. En este caso (en el que el valor es 1) $\hat{y}_t(1)$ y $Y_t(1)$ son los

valores asociados a la estima y covarianza de la información cuando no se ha incorporado todavía ninguna información sensorial.

▪ ***Incorporación de la evidencia sensorial y actualización del estado del sistema para todos los instantes de muestreo***

Supongamos que de repente tenemos disponible toda la información sensorial, por lo que podemos incorporarla en los potenciales sensoriales del árbol de unión para que ésta sea utilizada para actualizar el valor estimado del estado del sistema. De esta forma, el nuevo valor estimado para cada instante de muestreo estará en función de la información sensorial capturada en dicho instante y en todos los instantes previos. El orden que garantiza el mínimo número de operaciones del algoritmo de inferencia en este caso es el siguiente: inicialmente se incorpora la evidencia disponible en los conglomerados sensoriales del árbol $\mathbf{x}_t \mathbf{z}_{s,t}$, a continuación se realizan etapas de absorción de la evidencia de los nodos de transición $\mathbf{x}_{t-1} \mathbf{x}_t$ sobre sus nodos sensoriales adyacentes $\mathbf{x}_t \mathbf{z}_{s,t}$ (mensajes tipo (3) de la figura C2-4), y por último se realiza una absorción ordenada hacia delante entre nodos de transición adyacentes $\mathbf{x}_t \mathbf{x}_{t+1}$ (mensajes tipo (1) de la figura C2-4). Las operaciones de las tres etapas se presentan a continuación.

En la expresión (C2-22) se introduce la evidencia sensorial $\zeta_{s,t}$ existente para cada sensor e instante de muestreo en el potencial de su conglomerados correspondientes $\mathbf{x}_t \mathbf{z}_{s,t}$. El potencial deja de depender de la variable $\mathbf{z}_{s,t}$ para pasar a depender del valor de la medida $\zeta_{s,t}$. Tras estas operaciones los potenciales de los conglomerados sensoriales almacenan, según la nomenclatura del filtro de Información, su información sensorial $\mathbf{i}_{s,t}$ y su covarianza $\mathbf{I}_{s,t}$.

$$\Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^2(\mathbf{x}_t, \zeta_{s,t}) = \Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^1(\mathbf{x}_t, \mathbf{z}_{s,t} \mid \mathbf{z}_{s,t} = \zeta_{s,t}) = \left(g, \left[\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \zeta_{s,t} \right], \left[\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} \right] \right) = \left(g, \left[\mathbf{i}_{s,t} \right], \left[\mathbf{I}_{s,t} \right] \right) \quad (\text{C2.22})$$

con $t = 1:n, s = 1:S$

Después de introducir la información sensorial, se realizan las siguientes operaciones, asociadas a la absorción de los nodos de transición $\mathbf{x}_{t-1} \mathbf{x}_t$ sobre sus nodos sensoriales adyacentes $\mathbf{x}_t \mathbf{z}_{s,t}$. En la fórmula (C2-23) se marginaliza el potencial asociado al conglomerado $\mathbf{x}_t \mathbf{z}_{s,t}$ sobre la variable $\mathbf{z}_{s,t}$ y como dicho potencial ya no depende de dicha variable el potencial resultado de la marginalización almacena también la información sensorial $\mathbf{i}_{s,t}$ y su covarianza $\mathbf{I}_{s,t}$. La expresión (C2-24) es la actualización del valor de los potenciales de los separadores $\mathbf{x}_{t-1} \mathbf{x}_t$ con el resultado de la operación de marginalización. Por último, las expresiones (C2-25) y (C2-26) recogen las modificaciones de los potenciales de los conglomerados de transición $\mathbf{x}_{t-1} \mathbf{x}_t$ con la nueva información procedente de los potenciales de los separadores $\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1} \mathbf{x}_t$ adyacentes actualizados en (C2-24). Como en el caso multisensorial el número de separadores $\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1} \mathbf{x}_t$ es mayor que uno

en la operación (C2-26) aparece un cociente de productos de potenciales compuesto por tantos factores como nodos sensoriales $\mathbf{x}_t \mathbf{z}_{s,t}$ haya en el árbol de unión.

$$\Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^{2*}(\mathbf{x}_t, \zeta_{s,t}) = \int \Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^2(\mathbf{x}_t, \zeta_{s,t}) d\mathbf{z}_{s,t} = \Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^2(\mathbf{x}_t, \zeta_{s,t}) = (\mathcal{G}, [\mathbf{i}_{s,t}], [\mathbf{I}_{s,t}]) \quad \text{con } t = 1:n, s = 1:S \quad (\text{C2.23})$$

$$\Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1} \mathbf{x}_t}^2(\mathbf{x}_t, \zeta_{s,t}) = \Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^{2*}(\mathbf{x}_t, \zeta_{s,t}) = (\mathcal{G}, [\mathbf{i}_{s,t}], [\mathbf{I}_{s,t}]) \quad \text{con } t = 1:n, s = 1:S \quad (\text{C2.24})$$

$$\Phi_{\mathbf{x}_0 \mathbf{x}_t}^2(\mathbf{x}_0, \mathbf{x}_t, \zeta_{1:S,t}) = \Phi_{\mathbf{x}_0 \mathbf{x}_t}^1(\mathbf{x}_0, \mathbf{x}_t) \cdot \frac{\prod_s \Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_0 \mathbf{x}_t}^2(\mathbf{x}_t, \zeta_{s,t})}{\prod_s \Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_0 \mathbf{x}_t}^1(\mathbf{x}_t)} = \quad (\text{C2.25})$$

$$\begin{aligned} &= \left(\mathcal{G}, \begin{bmatrix} \hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \\ \mathbf{Q}_0^{-1} \mathbf{u}_0 \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 & -\mathbf{F}_0^T \mathbf{Q}_0^{-1} \\ -\mathbf{Q}_0^{-1} \mathbf{F}_0 & \mathbf{Q}_0^{-1} \end{bmatrix} \right) \cdot \frac{\prod_s (\mathcal{G}, [\mathbf{i}_{s,t}], [\mathbf{I}_{s,t}])}{\prod_s (\mathcal{O}, [\mathbf{0}], [\mathbf{0}])} = \\ &= \left(\mathcal{G}, \begin{bmatrix} \hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \\ \sum_s \mathbf{i}_{s,t} + \mathbf{Q}_0^{-1} \mathbf{u}_0 \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 & -\mathbf{F}_0^T \mathbf{Q}_0^{-1} \\ -\mathbf{Q}_0^{-1} \mathbf{F}_0 & \sum_s \mathbf{I}_{s,t} + \mathbf{Q}_0^{-1} \end{bmatrix} \right) \quad \text{con } s = 1:S \\ &\Phi_{\mathbf{x}_t \mathbf{x}_{t+1}}^2(\mathbf{x}_t, \mathbf{x}_{t+1}, \zeta_{1:S,t+1}) = \Phi_{\mathbf{x}_t \mathbf{x}_{t+1}}^1(\mathbf{x}_t, \mathbf{x}_{t+1}) \cdot \frac{\prod_s \Phi_{\mathbf{x}_{t+1} \mathbf{z}_{s,t+1} \cap \mathbf{x}_t \mathbf{x}_{t+1}}^2(\mathbf{x}_{t+1}, \zeta_{s,t+1})}{\prod_s \Phi_{\mathbf{x}_{t+1} \mathbf{z}_{s,t+1} \cap \mathbf{x}_t \mathbf{x}_{t+1}}^1(\mathbf{x}_{t+1})} = \quad (\text{C2.26}) \\ &= \left(\mathcal{G}, \begin{bmatrix} \hat{\mathbf{y}}_t(1) - \mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{u}_t \\ \mathbf{Q}_t^{-1} \mathbf{u}_t \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_t(1) + \mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{F}_t & -\mathbf{F}_t^T \mathbf{Q}_t^{-1} \\ -\mathbf{Q}_t^{-1} \mathbf{F}_t & \mathbf{Q}_t^{-1} \end{bmatrix} \right) \cdot \frac{\prod_s (\mathcal{G}, [\mathbf{i}_{s,t+1}], [\mathbf{I}_{s,t+1}])}{\prod_s (\mathcal{O}, [\mathbf{0}], [\mathbf{0}])} = \\ &= \left(\mathcal{G}, \begin{bmatrix} \hat{\mathbf{y}}_t(1) - \mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{u}_t \\ \sum_s \mathbf{i}_{s,t+1} + \mathbf{Q}_t^{-1} \mathbf{u}_t \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_t(1) + \mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{F}_t & -\mathbf{F}_t^T \mathbf{Q}_t^{-1} \\ -\mathbf{Q}_t^{-1} \mathbf{F}_t & \sum_s \mathbf{I}_{s,t+1} + \mathbf{Q}_t^{-1} \end{bmatrix} \right) \quad \text{con } t = 1:n-1, s = 1:S \end{aligned}$$

La diferencia existente entre las expresiones de los potenciales de los conglomerados de transición antes de incorporar en los mismos la evidencia sensorial para ese instante de muestreo (potenciales (1) y tipo (2) de la tabla C2-8) y después de hacerlo (potenciales de las expresiones (C2-25) y (C2-26)), se encuentra en la aparición, para el segundo de los casos, de un sumatorio sobre la información sensorial $\mathbf{i}_{s,t}$ y su covarianza $\mathbf{I}_{s,t}$. Además, hay que destacar que todos los potenciales actualizados en esta etapa de absorción recogen la probabilidad conjunta sobre las variables aleatorias de las que siguen dependiendo y sobre la evidencia sensorial propagada hasta dicho potencial.

Los nuevos valores de todos los potenciales tras las absorciones hacia arriba se recogen en la siguiente tabla. En ella se puede observar como los valores de los separadores existentes entre los conglomerados de transición $\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}$ no dependen todavía de ninguna evidencia sensorial, al no haberse propagado todavía hacia delante, y de forma ordenada, la evidencia entre los conglomerados de transición $\mathbf{x}_t \mathbf{x}_{t+1}$. Así mismo, los potenciales de los conglomerados de transición $\mathbf{x}_t \mathbf{x}_{t+1}$ únicamente dependen, en esta etapa del algoritmo y desde el punto de vista sensorial, de la evidencia asociada a su correspondiente instante de muestreo.

Árbol de conglomerados del problema	
Potenciales tras propagar la información sensorial hacia arriba	
(1)	$\Phi_{x_0x_1}^2(x_0, x_1, \zeta_{1:S,1}) = \Phi_{x_0x_1}^1(x_0, x_1) \cdot \frac{\prod_s \Phi_{x_1z_{s,1} \cap x_0x_1}^2(x_1, \zeta_{s,1})}{\prod_s \Phi_{x_1z_{s,1} \cap x_0x_1}^1(x_1)} =$ $= \left(g, \left[\hat{y}_0 - F_0^T Q_0^{-1} u_0, \sum_s i_{s,1} + Q_0^{-1} u_0 \right], \begin{bmatrix} Y_0 + F_0^T Q_0^{-1} F_0 & -F_0^T Q_0^{-1} \\ -Q_0^{-1} F_0 & \sum_s I_{s,1} + Q_0^{-1} \end{bmatrix} \right) \quad \text{con } s = 1 : S$
(2)	$\Phi_{x_t x_{t+1}}^2(x_t, x_{t+1}, \zeta_{t:S,t+1}) = \Phi_{x_t x_{t+1}}^1(x_t, x_{t+1}) \cdot \frac{\prod_s \Phi_{x_{t+1}z_{s,t+1} \cap x_t x_{t+1}}^2(x_{t+1}, \zeta_{s,t+1})}{\prod_s \Phi_{x_{t+1}z_{s,t+1} \cap x_t x_{t+1}}^1(x_{t+1})} =$ $= \left(g, \left[\hat{y}_t(1) - F_t^T Q_t^{-1} u_t, \sum_s i_{s,t+1} + Q_t^{-1} u_t \right], \begin{bmatrix} Y_t(1) + F_t^T Q_t^{-1} F_t & -F_t^T Q_t^{-1} \\ -Q_t^{-1} F_t & \sum_s I_{s,t+1} + Q_t^{-1} \end{bmatrix} \right) \quad \text{con } t = 1 : n-1, s = 1 : S$
(3)	$\Phi_{x_t z_{s,t}}^2(x_t, \zeta_{s,t}) = \Phi_{x_t z_{s,t}}^1(x_t, z_{s,t} z_{s,t} = \zeta_{s,t}) = (g, [i_{s,t}], [I_{s,t}]) = (g, [H_{s,t}^T R_{s,t}^{-1} \zeta_{s,t}], [H_{s,t}^T R_{s,t}^{-1} H_{s,t}])$ <p style="text-align: center;">con $t = 1 : n, s = 1 : S$</p>
(4)	$\Phi_{x_t z_{s,t} \cap x_{t-1} x_t}^2(x_t, \zeta_{s,t}) = \Phi_{x_t z_{s,t}}^{2*}(x_t, \zeta_{s,t}) = (g, [i_{s,t}], [I_{s,t}]) \quad \text{con } t = 1 : n, s = 1 : S$
(5)	$\Phi_{x_{t-1} x_t \cap x_t x_{t+1}}^2(x_t) = \Phi_{x_{t-1} x_t \cap x_t x_{t+1}}^1(x_t) = (g, [\hat{y}_t(1)], [Y_t(1)]) =$ $= \left(g, \left[(F_{t-1} Y_{t-1}^{-1}(1) F_{t-1}^T + Q_{t-1})^{-1} (F_{t-1} Y_{t-1}^{-1}(1) \hat{y}_{t-1}(1) + u_{t-1}) \right], \left[(F_{t-1} Y_{t-1}^{-1}(1) F_{t-1}^T + Q_{t-1})^{-1} \right] \right)$ <p style="text-align: center;">con $t = 1 : n-1, \hat{y}_0(1) = \hat{y}_0, Y_0(1) = Y_0$</p>

Tabla C2-9. Árbol de unión de la red de la tabla C2-6, con S=3 sensores, y sus potenciales asociados, tanto para los diferentes conglomerados (1, 2 y 3) como para los separadores (4 y 5), después de propagar hacia arriba la información de todos los sensores para todos los instantes de muestreo.

Por último, con el objetivo de actualizar el valor estimado de la información del estado del sistema para cada instante de muestreo, de forma que éste incorpore toda la información sensorial existente salvo la medidas tomadas para instantes de muestreo posteriores, hay que realizar la absorción hacia delante y de forma ordenada de los potenciales de los conglomerados de transición $x_t x_{t+1}$ (envío de mensajes tipo (1) de la figura C2-4).

La primera absorción se da entre los conglomerados de transición $x_0 x_1$ y $x_1 x_2$ y sus operaciones son recogidas en las siguientes expresiones. En (C2-27) se marginaliza el potencial asociado al conglomerado $x_0 x_1$ sobre la variable x_0 para obtener el valor del separador $x_0 x_1 \cap x_1 x_2$, que es actualizado en la expresión (C2-28). El cálculo del valor de la K^* y h^* de la expresión (C2-27) es análogo al cálculo de la K^* y h^* de la expresión (C2-19), ya que ambas expresiones únicamente se

diferencian por la existencia de los sumatorios asociados a la información sensorial para el caso (C2-27). En la expresión (C2-29) se actualiza el potencial del conglomerado $\mathbf{x}_1\mathbf{x}_2$ al añadirle la nueva información procedente del separador (en la que se encuentra el resultado de la predicción y asimilación sensorial) y quitarle la información almacenada previamente en el mismo (que contenía únicamente el resultado de la predicción sin información sensorial, al no estar ésta disponible en la primera etapa de absorción del caso hipotético propuesto).

$$\Phi_{\mathbf{x}_0\mathbf{x}_I}^{2*}(\mathbf{x}_I, \zeta_{1:S,I}) = \int \Phi_{\mathbf{x}_0\mathbf{x}_I}^2(\mathbf{x}_0, \mathbf{x}_I, \zeta_{1:S,I}) d\mathbf{x}_0 = (\mathbf{g}, \mathbf{h}^*, \mathbf{K}^*) = (\mathbf{g}, [\hat{\mathbf{y}}_I(3)], [\mathbf{Y}_I(3)]) \quad \text{con } s = 1 : S \quad (\text{C2.27})$$

$$\begin{aligned} \mathbf{K}^* &= \sum_s \mathbf{I}_{s,I} + \mathbf{Q}_0^{-1} - \mathbf{Q}_0^{-1}\mathbf{F}_0 \left(\hat{\mathbf{Y}}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 \right)^{-1} \mathbf{F}_0^T \mathbf{Q}_0^{-1} = \sum_s \mathbf{I}_{s,I} + \hat{\mathbf{Y}}_I(1) = \hat{\mathbf{Y}}_I(3) \\ \mathbf{h}^* &= \sum_s \mathbf{i}_{s,I} + \mathbf{Q}_0^{-1} \mathbf{u}_0 + \mathbf{Q}_0^{-1} \mathbf{F}_0 \left(\mathbf{Y}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 \right)^{-1} \left(\hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \right) = \sum_s \mathbf{i}_{s,I} + \hat{\mathbf{y}}_I(1) = \hat{\mathbf{y}}_I(3) \end{aligned}$$

$$\begin{aligned} \Phi_{\mathbf{x}_0\mathbf{x}_I \cap \mathbf{x}_I\mathbf{x}_2}^3(\mathbf{x}_I, \zeta_{1:S,I}) &= \Phi_{\mathbf{x}_0\mathbf{x}_I}^{2*}(\mathbf{x}_I, \zeta_{1:S,I}) = (\mathbf{g}, [\hat{\mathbf{y}}_I(3)], [\mathbf{Y}_I(3)]) = \\ &= \left(\mathbf{g}, \left[\sum_s \mathbf{i}_{s,I} + (\mathbf{F}_0 \mathbf{Y}_0^{-1} \mathbf{F}_0^T + \mathbf{Q}_0)^{-1} (\mathbf{F}_0 \mathbf{Y}_0^{-1} \hat{\mathbf{y}}_0 + \mathbf{u}_0) \right], \left[\sum_s \mathbf{I}_{s,I} + (\mathbf{F}_0 \mathbf{Y}_0^{-1} \mathbf{F}_0^T + \mathbf{Q}_0)^{-1} \right] \right) \quad \text{con } s = 1 : S \end{aligned} \quad (\text{C2.28})$$

$$\begin{aligned} \Phi_{\mathbf{x}_I\mathbf{x}_2}^3(\mathbf{x}_I, \mathbf{x}_2, \zeta_{1:S,I;2}) &= \Phi_{\mathbf{x}_I\mathbf{x}_2}^2(\mathbf{x}_I, \mathbf{x}_2, \zeta_{1:S,I;2}) \cdot \frac{\Phi_{\mathbf{x}_0\mathbf{x}_I \cap \mathbf{x}_I\mathbf{x}_2}^3(\mathbf{x}_I, \zeta_{1:S,I})}{\Phi_{\mathbf{x}_0\mathbf{x}_I \cap \mathbf{x}_I\mathbf{x}_2}^2(\mathbf{x}_I)} = \\ &= \left(\mathbf{g}, \left[\begin{array}{c} \hat{\mathbf{y}}_I(1) - \mathbf{F}_I^T \mathbf{Q}_I^{-1} \mathbf{u}_I \\ \sum_s \mathbf{i}_{s,2} + \mathbf{Q}_I^{-1} \mathbf{u}_I \end{array} \right], \left[\begin{array}{cc} \mathbf{Y}_I(1) + \mathbf{F}_I^T \mathbf{Q}_I^{-1} \mathbf{F}_I & -\mathbf{F}_I^T \mathbf{Q}_I^{-1} \\ -\mathbf{Q}_I^{-1} \mathbf{F}_I & \sum_s \mathbf{I}_{s,2} + \mathbf{Q}_I^{-1} \end{array} \right] \right) \cdot \frac{(\mathbf{g}, [\hat{\mathbf{y}}_I(3)], [\mathbf{Y}_I(3)])}{(\mathbf{g}, [\hat{\mathbf{y}}_I(1)], [\mathbf{Y}_I(1)])} = \\ &= \left(\mathbf{g}, \left[\begin{array}{c} \hat{\mathbf{y}}_I(3) - \mathbf{F}_I^T \mathbf{Q}_I^{-1} \mathbf{u}_I \\ \sum_s \mathbf{i}_{s,2} + \mathbf{Q}_I^{-1} \mathbf{u}_I \end{array} \right], \left[\begin{array}{cc} \mathbf{Y}_I(3) + \mathbf{F}_I^T \mathbf{Q}_I^{-1} \mathbf{F}_I & -\mathbf{F}_I^T \mathbf{Q}_I^{-1} \\ -\mathbf{Q}_I^{-1} \mathbf{F}_I & \sum_s \mathbf{I}_{s,2} + \mathbf{Q}_I^{-1} \end{array} \right] \right) \quad \text{con } s = 1 : S \end{aligned} \quad (\text{C2.29})$$

El nuevo valor del potencial del separador $\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{x}_2$, obtenido en la marginalización (C2-27) almacena el valor estimado para la información del estado $\hat{\mathbf{y}}_I(3)$, y su correspondiente covarianza $\mathbf{Y}_I(3)$, después de las etapas de predicción y asimilación sensorial del filtro de Información. Después de la absorción propuesta, en el potencial (C2-29) del segundo conglomerado de transición $\mathbf{x}_1\mathbf{x}_2$ aparece el resultado de la predicción y asimilación del filtro ($\hat{\mathbf{y}}_I(3)$ y $\mathbf{Y}_I(3)$) en vez del resultado de la predicción sin asimilación recogido en el potencial (1) de la tabla C2-9.

De la misma forma que ocurrió durante el primer proceso de absorción hacia delante y de forma ordenada, los resultados del resto de las absorciones se desprenden directamente de la similitud existente entre la fórmula (C2-29) y la expresión (1) de la tabla C2-9. Las expresiones de los potenciales asociados a los conglomerados y separadores después de finalizar el proceso de absorción hacia delante se presentan en la tabla C2-10.

Tras esta nueva etapa completa de inferencia, se puede observar que los potenciales ((1) y tipo (2) de la tabla C2-10) de los conglomerados de las transiciones $\mathbf{x}_t\mathbf{x}_{t+1}$ dependen, además de las variables propias de dicho potencial, de toda la información sensorial disponible $\zeta_{1:3,t}$ hasta el instante t ; que los potenciales (tipo (5)) de los separadores que hay entre los conglomerados de

transición $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}$ almacenan los mismos valores que obtendría el filtro de Información distribuido para la información del estado $\hat{\mathbf{y}}_t(3)$ y su covarianza $\mathbf{Y}_t(3)$. Además, el resto de los potenciales (tipo (3) y tipo (4)) mantiene el valor asignado durante el segundo proceso de absorción, en el que se introducía y propagaba la evidencia desde los conglomerados sensoriales $\mathbf{x}_t\mathbf{z}_{s,t}$ hacia los conglomerados de transición $\mathbf{x}_t\mathbf{x}_{t+1}$.

Árbol de conglomerados del problema	
Potenciales tras propagar información sensorial hacia delante	
(1)	$\Phi_{\mathbf{x}_0\mathbf{x}_1}^3(\mathbf{x}_0, \mathbf{x}_1, \zeta_{1:S,1}) = \Phi_{\mathbf{x}_0\mathbf{x}_1}^2(\mathbf{x}_0, \mathbf{x}_1, \zeta_{1:S,1}) =$ $= \left(g, \left[\hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0, \sum_s \mathbf{i}_{s,1} + \mathbf{Q}_0^{-1} \mathbf{u}_0 \right], \begin{bmatrix} \mathbf{Y}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 & -\mathbf{F}_0^T \mathbf{Q}_0^{-1} \\ -\mathbf{Q}_0^{-1} \mathbf{F}_0 & \sum_s \mathbf{I}_{s,1} + \mathbf{Q}_0^{-1} \end{bmatrix} \right) \quad \text{con } s = 1 : S$
(2)	$\Phi_{\mathbf{x}_t\mathbf{x}_{t+1}}^3(\mathbf{x}_t, \mathbf{x}_{t+1}, \zeta_{1:S,t:t+1}) = \Phi_{\mathbf{x}_t\mathbf{x}_{t+1}}^2(\mathbf{x}_t, \mathbf{x}_{t+1}, \zeta_{1:S,t:t+1}) \cdot \frac{\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}^3(\mathbf{x}_t, \zeta_{1:S,t:t})}{\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}^2(\mathbf{x}_t)} =$ $= \left(g, \left[\hat{\mathbf{y}}_t(3) - \mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{u}_t, \sum_s \mathbf{i}_{s,t+1} + \mathbf{Q}_t^{-1} \mathbf{u}_t \right], \begin{bmatrix} \mathbf{Y}_t(3) + \mathbf{F}_t^T \mathbf{Q}_t^{-1} \mathbf{F}_t & -\mathbf{F}_t^T \mathbf{Q}_t^{-1} \\ -\mathbf{Q}_t^{-1} \mathbf{F}_t & \sum_s \mathbf{I}_{s,t+1} + \mathbf{Q}_t^{-1} \end{bmatrix} \right) \quad \text{con } t = 1 : n-1, s = 1 : S$
(3)	$\Phi_{\mathbf{x}_t\mathbf{z}_{s,t}}^3(\mathbf{x}_t, \zeta_{s,t}) = \Phi_{\mathbf{x}_t\mathbf{z}_{s,t}}^2(\mathbf{x}_t, \zeta_{s,t}) = (g, [\mathbf{i}_{s,t}], [\mathbf{I}_{s,t}]) = (g, [\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \zeta_{s,t}], [\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t}])$ <p style="text-align: center;">con $t = 1 : n, s = 1 : S$</p>
(4)	$\Phi_{\mathbf{x}_t\mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}^3(\mathbf{x}_t, \zeta_{s,t}) = \Phi_{\mathbf{x}_t\mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}^2(\mathbf{x}_t, \zeta_{s,t}) = (g, [\mathbf{i}_{s,t}], [\mathbf{I}_{s,t}]) \quad \text{con } t = 1 : n, s = 1 : S$
(5)	$\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}^3(\mathbf{x}_t, \zeta_{1:S,t:t}) = \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}^{2*}(\mathbf{x}_t, \zeta_{1:S,t:t}) = (g, [\hat{\mathbf{y}}_t(3)], [\mathbf{Y}_t(3)])$ $\hat{\mathbf{y}}_t(3) = \sum_s \mathbf{i}_{s,t} + (\mathbf{F}_{t-1} \mathbf{Y}_{t-1}^{-1}(3) \mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})^{-1} (\mathbf{F}_{t-1} \mathbf{Y}_{t-1}^{-1}(3) \hat{\mathbf{y}}_{t-1}(3) + \mathbf{u}_{t-1})$ $\mathbf{Y}_t(3) = \sum_s \mathbf{I}_{s,t} + (\mathbf{F}_{t-1} \mathbf{Y}_{t-1}^{-1}(3) \mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})^{-1}$ <p style="text-align: center;">con $t = 1 : n-1, s = 1 : S, \hat{\mathbf{y}}_0(3) = \hat{\mathbf{y}}_0 \quad e \quad \mathbf{Y}_0(3) = \mathbf{Y}_0$</p>

Tabla C2-10. Árbol de unión de la red de la tabla C2-6, con $S=3$ sensores, y sus potenciales asociados, tanto para los diferentes conglomerados (1, 2 y 3) como para los separadores (4 y 5), después de propagar hacia delante y de forma ordenada toda la información sensorial

Por lo tanto, se ha demostrado la equivalencia entre el algoritmo de inferencia propuesto para el caso hipotético en el que se incorporan las medidas después de estimar el valor del estado sin las mismas, y el filtro de Información distribuido en el que las medidas se incorporan, de forma ordenada y secuencial, en cada una de sus etapas de predicción-asimilación.

▪ **Generalización del algoritmo de inferencia del ejemplo para el caso en el que las medidas llegan al algoritmo de forma temporalmente desordenada**

Por último vamos a presentar el algoritmo para un caso general en el que: (1) las medidas tomadas en el instante t no sólo puedan estar disponibles después de que ya se ha estimado la información del estado (y su covarianza) para ese instante, sino que unas medidas puedan llegar antes que otras, y (2) se desee obtener en todo momento una estima del estado del sistema que incorpore toda la información disponible en cada instante.

Dicho algoritmo se fundamentará en las siguientes observaciones, extraídas de las propiedades del algoritmo de inferencia exacto utilizado y del análisis de las operaciones realizadas para el caso hipotético de la sección anterior:

- Existe un conglomerado sensorial donde introducir como evidencia cada una de las posibles medidas, y el hecho de introducir la información sensorial en su conglomerado correspondiente (C2-22) únicamente modifica el valor del potencial de dicho conglomerado de forma que éste almacene la información sensorial en la forma del filtro de Información ($\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$). Por lo tanto, la información sensorial se puede introducir en cualquier momento del algoritmo en el potencial $\mathbf{x}_t \mathbf{z}_{s,t}$ asociado al sensor s e instante t en el que se realizó la medida $\zeta_{s,t}$.
- No es necesario realizar simultáneamente todas las absorciones tipo (3) correspondientes a un nodo de transición de estados, sino que éstas se pueden realizar en etapas diferentes del algoritmo, para evitar que éstas absorciones sean realizadas antes de que se haya introducido la evidencia en el potencial asociado a cada conglomerado sensorial. Por cada absorción se actualizará: (1) el potencial asociado al separador existente entre el nodo de transición y el de medida (C2-24), y (2) el potencial de transición multiplicándolo por el nuevo valor del potencial del separador (cuyos valores \mathbf{h} y \mathbf{K} almacenan respectivamente la información sensorial $\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$ en la forma del filtro de Información) y dividiéndolo entre el antiguo (cuyos valores \mathbf{h} y \mathbf{K} se encuentran inicializados a 0). Como resultado de cada absorción sensorial, $\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$ serán añadidos, respectivamente y como términos aditivos, a la segunda y cuarta componente del vector \mathbf{h} y la matriz \mathbf{K} del potencial asociado a la transición de estados. Los sumatorios de las expresiones (C2-25) y (C2-26) serán el resultado de las absorciones producidas sobre cada nodo sensorial una vez que todas las medidas de un instante de muestreo se encuentren disponibles.
- Los potenciales (1) y tipo (2) de la tabla C2-10 de los conglomerados de las transiciones $\mathbf{x}_t \mathbf{x}_{t+1}$ almacenan en componentes distintas de su vector \mathbf{h} y matriz \mathbf{K} la información sensorial $\zeta_{1:s,1:t-1}$ previa a t (recogida en la información del estado $\hat{\mathbf{y}}_t(3)$ y su covarianza $\mathbf{Y}_t(3)$) y la información sensorial $\zeta_{1:s,t}$ del instante t . Esta división es fundamental para el

buen funcionamiento del algoritmo en el caso en el que llega la información retardada (como en el ejemplo hipotético propuesto en la sección anterior) ya que es lo que hace que la marginalización de la expresión (C2-27) y la marginalización de la expresión (C2-19) únicamente se diferencien respectivamente en la existencia/ausencia del termino sumatorio relacionado con la información sensorial. Cuando la información sensorial además de retardada también se introduce de forma desordenada, la separación de la información sensorial previa a t y del instante t se mantendrá y el sumatorio de la expresión de marginalización (C2-27) en vez de contener toda la información sensorial en el instante t contendrá toda la información sensorial del instante t que se encuentre disponible en esa etapa del algoritmo de inferencia.

- Por último, es importante también destacar que cada vez que se introduce una medida $\zeta_{s,t}$ y se propaga la evidencia hacia su conglomerado de transición $\mathbf{x}_{t-1}\mathbf{x}_t$, no es necesario realizar una etapa completa y ordenada de absorciones tipo (1) para actualizar la información del estado del sistema para todo instante de muestreo, ya que la información sensorial en el instante t no va a modificar la información del estado en instantes anteriores. Por lo tanto, únicamente es necesario realizar una etapa ordenada de absorciones tipo (1) que comiencen en el conglomerado de transición $\mathbf{x}_{t-1}\mathbf{x}_t$ asociado al instante de medida, por lo que la primera absorción que habrá que realizar será $\mathbf{x}_{t-1}\mathbf{x}_t \rightarrow \mathbf{x}_t\mathbf{x}_{t+1}$.

Todas estas propiedades soportan el correcto funcionamiento del siguiente algoritmo de inferencia para medidas asíncronas, retardadas y temporalmente desordenadas y su equivalencia con el filtro de Información distribuido y asíncrono (con medidas temporalmente ordenadas y no retardadas):

- 1) Inicialización
 - a. Obtener el modelo del sistema (ecuación de transición de estados) y de los sensores (ecuaciones sensoriales).
 - b. Construir la red bayesiana, con tantos nodos sensoriales como sean necesarios y su árbol de unión correspondiente.
 - c. Convertir las ecuaciones probabilísticas de transición de estado y sensoriales en potenciales condicionales.
 - d. Inicializar los potenciales de los conglomerados (a partir de los potenciales condicionales) y de los separadores (a 0).
 - e. Realizar absorciones tipo (1) de forma sucesiva y ordenada de izquierda a derecha desde el primer nodo de transición de estados hasta el último (primera absorción $\mathbf{x}_0\mathbf{x}_1 \rightarrow \mathbf{x}_1\mathbf{x}_2$, siguiente absorción $\mathbf{x}_1\mathbf{x}_2 \rightarrow \mathbf{x}_2\mathbf{x}_3$, ... , y última absorción $\mathbf{x}_{n-2}\mathbf{x}_{n-1} \rightarrow \mathbf{x}_{n-1}\mathbf{x}_n$)
- 2) Bucle principal: **while(true)**
 - a. Esperar hasta que haya una medida disponible $\zeta_{s,t}$ para cualquier t y s
 - b. Introducir la medida $\zeta_{s,t}$ como evidencia en el conglomerado sensorial $\mathbf{x}_t\mathbf{z}_{s,t}$
 - c. Realizar una absorción tipo (3) desde el nodo de transición de estados asociado al conglomeradonodo sensorial sobre el que se ha medido (absorción $\mathbf{x}_t\mathbf{z}_{s,t} \rightarrow \mathbf{x}_{t-1}\mathbf{x}_t$)

C2.2. Redes bayesianas continuas y fusión multisensorial, asíncrona y temporalmente desordenada

d. Si $t \neq n$, realizar absorciones tipo (1) de forma sucesiva y ordenada de izquierda a derecha desde el nodo de transición de estados asociado a la medida disponible hasta el último (primera absorción $\mathbf{x}_{t-1}\mathbf{x}_t \rightarrow \mathbf{x}_t\mathbf{x}_{t+1}$, ... , y última absorción $\mathbf{x}_{n-2}\mathbf{x}_{n-1} \rightarrow \mathbf{x}_{n-1}\mathbf{x}_n$)

Algoritmo C2-1. Algoritmo de inferencia exacto para la red bayesiana de la tabla C2-6 en el caso en el que las medidas puedan ser asíncronas, retardadas y estar temporalmente desordenadas.

Tras el proceso de inicialización, en los potenciales de los separadores entre conglomerados de transición se almacena el valor estimado, a partir del estado inicial y las señales de control, de la información y covarianza del estado del sistema para los diferentes instantes de muestreo. Cada vez que una nueva medida está disponible, tras un tiempo de procesamiento y/o de comunicación, ésta es introducida como evidencia (2.b), absorbida por el conglomerado de transición asociado a la medida (2.c) y utilizada para actualizar la estima de todos los estados del sistema no anteriores al instante de medida (2.d).

Para implementar el algoritmo propuesto basta con: (1) implementar las operaciones de la tabla C2-2 y las operaciones del proceso de absorción (ecuaciones (C2-1),(C2-2),(C2-3) y (C2-4)), (2) reservar memoria para las variables potenciales de tipo canónico para todos los conglomerados y separadores necesarios del árbol de unión asociado al sistema y sensores disponibles, y (3) escribir un programa que implemente sobre las variables de los potenciales las etapas propuestas en el algoritmo C2-1. El valor estimado del estado del sistema $\hat{\mathbf{x}}_t$ para cualquier instante de muestreo t se puede obtener del potencial asociado al separador $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}$, que almacenará en todo momento el valor estimado, con la información sensorial disponible por el algoritmo en ese instante, de la información del estado $\hat{\mathbf{y}}_t = \mathbf{Y}_t\hat{\mathbf{x}}_t$, y su correspondiente covarianza $\mathbf{Y}_t = \mathbf{P}_t^{-1}$.

C2.2.3. Algoritmo de inferencia exacta con una etapa de validación de medidas

Tanto el filtro de Kalman como el de Información pueden incluir una etapa de validación de medidas cuyo objeto es seleccionar que medidas se pueden utilizar para mejorar la estima del estado del sistema y cuales deben ser rechazadas y no incluidas en la etapa de asimilación porque se consideran erróneas. Considerando que estamos trabajando en un entorno probabilístico en el que no sólo se utiliza la información sensorial si no que además se modela y hace uso del error asociado a la misma, se podría optar por cuantificar en el modelo todas las posibles causas de error y eliminar de este modo la etapa de validación. Sin embargo, esa solución no es óptima para algunos tipos de errores como son los esporádicos o los debidos al mal funcionamiento del sensor, y no es factible cuando haya errores que no se puedan modelar.

En este apartado se analizan diferentes aspectos relacionados con la inclusión de la etapa de validación en el algoritmo de inferencia exacta propuesto en la sección anterior (algoritmo C2-1). Inicialmente, se presenta una nueva red bayesiana que modela e incorpora la etapa de validación de forma natural, y se proponen dos algoritmos de inferencia para el caso en el que se puede determinar de forma exacta la validez de la medida antes de que ésta sea incorporada como evidencia en el algoritmo. A continuación se analiza dentro del marco de las redes bayesianas los tests de validación basados en el valor estimado de la medida y se presenta una nueva versión del algoritmo de inferencia que incorpora los cálculos necesarios para estimar dicho valor. Por último, se resaltan los problemas existentes con la etapa de validación cuando las medidas llegan al algoritmo de forma temporalmente desordenada.

▪ **Modelado de la etapa de validación como parte de la red bayesiana**

La existencia de la etapa de validación puede ser fácilmente incluida en la red bayesiana propuesta para modelar un sistema lineal de control sobre el que varios sensores pueden realizar medidas de forma acíclica y temporalmente retardada (tabla C2-6), por medio de la incorporación de un nodo discreto $\mathbf{v}_{s,t}$ en la relación existente, para cada instante de muestreo \mathbf{t} , entre cada nodo sensorial $\mathbf{z}_{s,t}$ y nodo de estado \mathbf{x}_t . Además, es necesario sustituir la expresión de probabilidad condicional $\Pr(\mathbf{z}_{s,t} | \mathbf{x}_t)$ por una nueva expresión que capture la nueva relación existente entre las tres variables. En la parte superior de la tabla C2-11 se representan las modificaciones necesarias en la estructura de la red y en la expresión de probabilidad condicional para un nodo sensorial \mathbf{s} e instante de muestreo \mathbf{t} . Dicha modificación debe ser incluida en todos los instantes de muestreo en los sensores para los que se desee realizar una etapa de validación.

El nodo discreto de validación $\mathbf{v}_{s,t}$ tiene dos posibles estados (**true** y **false**) cada uno de los cuales establece una relación diferente entre el valor del sensor y el valor del estado. Dichas relaciones se encuentran representadas en la parte inferior de la tabla C2-11:

- En el caso en el que la medida se considera válida ($\mathbf{v}_{s,t}=\mathbf{true}$) se mantiene la relación propuesta para en la sección anterior, donde existía entre la medida y el estado una relación lineal no determinista, que incluía un ruido blanco aditivo.
- Para medidas no válidas ($\mathbf{v}_{s,t}=\mathbf{false}$) se eliminará la relación de dependencia existente entre el valor de la medida y el valor del estado para que la incorporación de medidas no válidas en la red no modifiquen el valor estimado del estado del sistema. Es decir, hay que elegir una función de probabilidad que cumpla que $\Pr(\mathbf{z}_{s,t} | \mathbf{x}_t, \mathbf{v}_{s,t} = \mathbf{false}) = \Pr(\mathbf{z}_{s,t} | \mathbf{v}_{s,t} = \mathbf{false})$. Aunque en principio se puede elegir una función de probabilidad cualquiera que cumpla esa propiedad, ya que ésta implica que el conocimiento del valor de la medida $\mathbf{z}_{s,t}$ no modificará el valor estimado del estado \mathbf{x}_t , el algoritmo de inferencia elegido únicamente trabaja con

funciones de probabilidad condicional gaussiana multivariable. Por lo tanto, utilizaremos una función de este tipo que cumpla la condición de independencia condicional. La función elegida es $N_{z_{s,t}}(\mu_{z_{s,t}}, \mathbf{R}_{s,t})$, y en ella se supone que el valor medio de la medida $\mu_{z_{s,t}}$ tomará un valor desconocido a priori, lo que puede ser interpretado como la existencia de una dependencia de la medida del sensor de una variable desconocida.

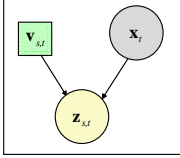
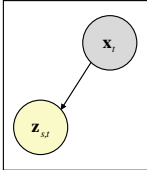
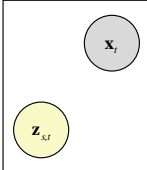
Inclusión de un nodo de validación discreto en la relación existente entre el nodo sensor y el nodo estado (para todo s y todo t)	
	$\Pr(\mathbf{z}_{s,t} \mathbf{x}_t, \mathbf{v}_{s,t}) :$ $\Pr(\mathbf{z}_{s,t} \mathbf{x}_t, \mathbf{v}_{s,t} = \text{true}) = N_{z_{s,t}}(\mathbf{0} + \mathbf{H}_{s,t} \mathbf{x}_t, \mathbf{R}_{s,t})$ $\Pr(\mathbf{z}_{s,t} \mathbf{x}_t, \mathbf{v}_{s,t} = \text{false}) = \Pr(\mathbf{z}_{s,t} \mathbf{v}_{s,t} = \text{false}) = N_{z_{s,t}}(\mu_{z_{s,t}}, \mathbf{R}_{s,t})$
a) Medidas válidas:  $\Pr(\mathbf{z}_{s,t} \mathbf{x}_t, \mathbf{v}_{s,t} = \text{true}) = N_{z_{s,t}}(\mathbf{0} + \mathbf{H}_{s,t} \mathbf{x}_t, \mathbf{R}_{s,t})$	b) Medidas no válidas:  $\Pr(\mathbf{z}_{s,t} \mathbf{x}_t, \mathbf{v}_{s,t} = \text{false}) = \Pr(\mathbf{z}_{s,t} \mathbf{v}_{s,t} = \text{false}) = N_{z_{s,t}}(\mu_{z_{s,t}}, \mathbf{R}_{s,t})$

Tabla C2-11. Incorporación de un nodo discreto en la relación existente entre cada nodo sensorial y el nodo de estado del sistema para un instante de tiempo y sensor determinado, con el objetivo de modelar la etapa de validación dentro del ámbito de las redes bayesianas.

La nueva red bayesiana (que incorpora, para cada sensor validable s e instante de muestreo t, un nodo de validación $\mathbf{v}_{s,t}$) es una red *mixta*, con nodos continuos y discretos, por lo que, en principio, no se puede utilizar ni el procedimiento explicado en el primer apartado de este capítulo para obtener el árbol de unión asociada a la misma, ni utilizar las operaciones con potenciales canónicos continuos propuestas en dicha sección.

Sin embargo, en el caso en el que se conozca el estado de la variable de validación $\mathbf{v}_{s,t}$ antes de incorporar la medida como evidencia en la red, i.e. se pueda determinar completamente la validez de la medida, podemos utilizar las operaciones sobre redes puramente continuas al considerar que tenemos una red continua con dos posibles potenciales para cada conglomerado sensorial, y utilizar el potencial adecuado después de conocer si la medida ha superado el test de validación con éxito o debe ser rechazada. En la siguiente ecuación se muestra el valor de los dos potenciales después del proceso de inicialización.

$$\Phi_{\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t}}^0(\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t}) = \begin{cases} \text{si } \mathbf{v}_{s,t} = \text{true} \Rightarrow \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & -\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right) \\ \text{si } \mathbf{v}_{s,t} = \text{false} \Rightarrow \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{R}_{s,t}^{-1} \mu_{z_{s,t}} \end{bmatrix}, \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right) \end{cases} \quad (\text{C2.30})$$

En la sección anterior se demostró, para el caso en el que la matriz de medida era $\mathbf{H}_{s,t}$, que la información sensorial añadida a la información del estado durante el proceso de inferencia valía

$$\mathbf{i}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \zeta_{s,t} \text{ e } \mathbf{I}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t}.$$

La información sensorial ($\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$) para el caso no válido se obtiene introduciendo como evidencia en el potencial no válido el valor de la medida $\zeta_{s,t}$. Esta operación se encuentra recogida en la ecuación (C2-31), y como cabría esperar debido a la falta de relación de dependencia entre la variable \mathbf{x}_t y $\mathbf{z}_{s,t}$, el resultado de esta operación es una información sensorial relativa el estado nula ($\mathbf{i}_{s,t} = \mathbf{0}$ e $\mathbf{I}_{s,t} = \mathbf{0}$).

$$\Phi_{\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t}}^1(\mathbf{x}_t, \zeta_{s,t}, \mathbf{v}_{s,t} = \text{false}) = \Phi_{\mathbf{x}_t, \mathbf{z}_{s,t}}^0(\mathbf{x}_t, \mathbf{z}_{s,t} \mid \mathbf{z}_{s,t} = \zeta_{s,t}, \mathbf{v}_{s,t} = \text{false}) = (g, [\mathbf{0}], [\mathbf{0}]) = (g, [\mathbf{i}_{s,t}], [\mathbf{I}_{s,t}]) \quad (\text{C2.31})$$

Aun más, es importante observar que el valor medio de la medida $\mu_{z_{s,t}}$, desconocido a priori, no aparece de forma explícita en el potencial resultado de introducir la medida como evidencia, y por lo tanto, podría parecer que cualquier valor de $\mu_{z_{s,t}}$ resultará apropiado para realizar los cálculos. Esto no es realmente así, ya que el valor elegido de $\mu_{z_{s,t}}$ influye en el cálculo de la g del potencial de medida durante la inicialización y el valor de la medida $\zeta_{s,t}$ influye en el cálculo de la g del potencial de medida cuando se introduce la medida como evidencia. El cálculo de la g ha sido obviado durante el desarrollo de todas las fórmulas de este capítulo, ya que únicamente contiene información sobre la constante de normalización de los potenciales, y no sobre los estados y su covarianza.

Por lo tanto, siguiendo con la filosofía del resto del capítulo, obviaremos el cálculo de la g , y utilizaremos como valor inicial medio de medida $\mu_{z_{s,t}}$ para el caso no válido aquel que nos permita simplificar los cálculos y reducir la memoria necesaria para almacenar los potenciales. Una forma de lograr estos dos objetivos de forma conjunta se obtiene:

- Igualando el valor medio de medida $\mu_{z_{s,t}}$ a $\mathbf{0}$.
- Considerando que tenemos una única función de probabilidad condicional, representado en la ecuación (C2-32), y por lo tanto un único potencial para el conglomerado de medida $\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t}$, recogido en la ecuación (C2-33), y que ambas expresiones dependen de una matriz de medida $\mathbf{M}_{s,t}$ con dos posibles valores. Cuando la medida es válida ($\mathbf{v}_{s,t} = \text{true}$) la matriz $\mathbf{M}_{s,t}$ toma el valor del modelo de medida $\mathbf{H}_{s,t}$ y cuando no lo es ($\mathbf{v}_{s,t} = \text{false}$) la matriz es igual a $\mathbf{0}$.

$$\Pr(\mathbf{z}_{s,t} \mid \mathbf{x}_t, \mathbf{v}_{s,t}) = N_{z_{s,t}}(\mathbf{0} + \mathbf{M}_{s,t} \mathbf{x}_t, \mathbf{R}_{s,t}) \quad (\text{C2.32})$$

$$\text{con } t = 1 : n, s = 1 : S, \mathbf{M}_{s,t} = \begin{cases} \mathbf{H}_{s,t} & \text{si } \mathbf{v}_{s,t} = \text{true} \\ \mathbf{0} & \text{si } \mathbf{v}_{s,t} = \text{false} \end{cases}$$

$$\Phi_{\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t}}^0(\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t}) = \left(\mathcal{G}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{M}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{M}_{s,t} & -\mathbf{M}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{M}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right) \quad (\text{C2.33})$$

$$\text{con } t = 1 : n, s = 1 : S, \mathbf{M}_{s,t} = \begin{cases} \mathbf{H}_{s,t} & \text{si } \mathbf{v}_{s,t} = \text{true} \\ \mathbf{0} & \text{si } \mathbf{v}_{s,t} = \text{false} \end{cases}$$

- Suponiendo que durante el proceso de inicialización el potencial de medida $\mathbf{x}_t \mathbf{z}_{s,t}$ tiene como matriz de medida $\mathbf{H}_{s,t}$, ya que es la opción más adecuada cuando se considera que las medidas son habitualmente válidas. Una vez realizada la validación, si la medida es rechazada se modificará el potencial de medida igualando a $\mathbf{0}$ los términos del mismo que dependan de la matriz de medida $\mathbf{M}_{s,t}$. De esta forma, se romperá directamente la dependencia existente entre el estado y la media y una absorción tipo (3) transmitirá información sensorial nula relativa a los términos $\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$.

El algoritmo C2-2 recoge las modificaciones que hay que realizar sobre el algoritmo C2-1 para incluir la etapa de validación con este procedimiento. En la etapa de inicialización no se ha realizado ninguna modificación porque se supone que la mayoría de las medidas van a ser válidas, y como ya se ha mencionado, la inicialización con la matriz del modelo $\mathbf{H}_{s,t}$ resultará ventajosa. En el bucle principal, después de cada medida $\zeta_{s,t}$, se comprueba si ésta es válida. Cuando no lo es, se re-inicializa el valor del potencial asociado al conglomerado $\mathbf{x}_t \mathbf{z}_{s,t}$ haciendo $\mathbf{0}$ las componentes de la matriz \mathbf{K} de la tripleta del conglomerado dependientes del valor de $\mathbf{M}_{s,t}$. La nueva versión del algoritmo de inferencia se presenta a continuación.

```

1) Inicialización: realización de etapa 1) del algoritmo C2-1
2) Bucle principal: while(true)
    a. Esperar hasta que haya una medida disponible  $\zeta_{s,t}$  para cualquier  $t$  y  $s$ 
    b. Si la medida  $\zeta_{s,t}$  no es válida, modificar el potencial asociado a dicha medida  $\Phi(\mathbf{x}_t, \mathbf{z}_{s,t})$  igualando a  $\mathbf{0}$  las submatrices de  $\mathbf{K}$  relacionadas con  $\mathbf{x}_t$  (ver expresión (C2-33))
    c. Introducir la medida  $\zeta_{s,t}$  como evidencia en el conglomerado sensorial  $\mathbf{x}_t \mathbf{z}_{s,t}$ 
    d. Realizar una absorción tipo (3) desde el conglomerado de transición de estados asociado al conglomerado sensorial sobre el que se ha medido (absorción  $\mathbf{x}_t \mathbf{z}_{s,t} \rightarrow \mathbf{x}_{t-1} \mathbf{x}_t$ )
    e. Si  $t \neq n$ , realizar absorciones tipo (1) de forma sucesiva y ordenada de izquierda a derecha desde el nodo de transición de estados asociado a la medida disponible hasta el último (primera absorción  $\mathbf{x}_{t-1} \mathbf{x}_t \rightarrow \mathbf{x}_t \mathbf{x}_{t+1}$ , ... , y última absorción  $\mathbf{x}_{n-2} \mathbf{x}_{n-1} \rightarrow \mathbf{x}_{n-1} \mathbf{x}_n$ )

```

Algoritmo C2-2. Algoritmo de inferencia exacto sobre la red bayesiana que modela un sistema lineal sobre el que un grupo de sensores pueden proporcionar medidas no siempre válidas de forma asíncrona, retardada y temporalmente desordenada, para el caso en el que se determina si la medida es válida o no antes de introducirla como evidencia.

Aunque este modo de proceder trata de forma semejante toda la información sensorial y únicamente utiliza una matriz $\mathbf{M}_{s,t}$ diferente para medidas válidas y no válidas, en el caso de tener una medida no válida los cálculos debidos a la incorporación de la evidencia (etapa 2.c del algoritmo C2-2), a la absorción tipo (3) desde el nodo de transición de estados asociado a dicha medida (etapa 2.d), y a las absorciones sucesivas tipo (1) entre los nodos de transición de estados (etapa 3.e), sobrecargan computacionalmente el algoritmo sin producir ningún cambio en los potenciales de transición de estados sobre los que se han realizado las diferentes absorciones. Por lo tanto, para las medidas no válidas esas etapas no son realmente necesarias y únicamente consumen tiempo de procesamiento. Resulta mucho más sencillo y menos costoso computacionalmente modificar el algoritmo de inferencia C2-1, introduciendo una instrucción condicional antes de incorporar la evidencia, de forma que solo se realicen las etapas asociadas a dicha incorporación para medidas válidas. Aunque podría parecer que en esta solución, presentada en la siguiente versión del algoritmo, la etapa de validación deja de estar modelada como parte de la red bayesiana, esto no es así ya que no hemos modificado la red bayesiana si no que sólo hemos implementado un algoritmo de inferencia que minimiza el número de operaciones para realizar únicamente aquellas realmente necesarias. Además, una ventaja añadida a esta solución es que nos permite no hacer ninguna suposición sobre el tipo de función de probabilidad $\Pr(\mathbf{z}_{s,t} \mid \mathbf{v}_{s,t} = \text{false})$ que mejor modela el comportamiento de las medidas no válidas.

```

1) Inicialización: realización de etapa 1) del algoritmo C2-1
2) Bucle principal: while(true)
  a. Esperar hasta que haya una medida disponible  $\mathbf{z}_{s,t}$  para
    cualquier  $\mathbf{t}$  y  $\mathbf{s}$ 
  b. Si la medida es válida
    i. Introducir la medida  $\mathbf{z}_{s,t}$  como evidencia en el
       conglomerado sensorial  $\mathbf{x}_t \mathbf{z}_{s,t}$ 
    ii. Realizar una absorción tipo (3) desde el conglomerado de
        transición de estados asociado al conglomerado sensorial
        sobre el que se ha medido (absorción  $\mathbf{x}_t \mathbf{z}_{s,t} \rightarrow \mathbf{x}_{t-1} \mathbf{x}_t$ )
    iii. Si  $\mathbf{t} \neq \mathbf{n}$ , realizar absorciones tipo (1) de forma sucesiva
        y ordenada de izquierda a derecha desde el nodo de
        transición de estados asociado a la medida disponible
        hasta el último (primera absorción
         $\mathbf{x}_{t-1} \mathbf{x}_t \rightarrow \mathbf{x}_t \mathbf{x}_{t+1}$ , ... , y última absorción  $\mathbf{x}_{n-2} \mathbf{x}_{n-1} \rightarrow \mathbf{x}_{n-1} \mathbf{x}_n$ )

```

Algoritmo C2-3. Algoritmo de inferencia exacto para la red bayesiana de la tabla C2-6 en el caso en el que las medidas puedan ser no válidas, asíncronas, retardadas y estar temporalmente desordenadas y se incluya la etapa de validación con una instrucción condicional.

Para los casos en los que no se pueda determinar exactamente la validez de la medida, será necesario utilizar el algoritmo *mixto* de inferencia bayesiana, e incluir una probabilidad a priori en el nodo de validación $\mathbf{v}_{s,t}$ y/o enlazar ese nodo con otros nodos de la red que nos permitan estimar, de alguna forma, la probabilidad de que la medida sea válida. El algoritmo mixto de propagación utilizará la probabilidad de validez asignada a la variable $\mathbf{v}_{s,t}$ para obtener mediante

marginalización débil [Lauritzen1992] los nuevos valores para la información sensorial $\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$ que se que añadirán después de la absorción tipo (3) a la información sobre la estima del estado del sensor $\hat{\mathbf{y}}_t$ e \mathbf{Y}_t . El cálculo de los valores $\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$ se puede entender como el resultado de utilizar dos filtros de información durante la etapa de asimilación (uno en el que se introduce la medida y otro en el que no), y mezclar en base a la probabilidad de que la medida sea válida la información asociada a cada uno de los filtros. De esta forma la información sensorial influirá más en la información sobre el estado del sistema cuanto más probabilísticamente válida sea considerada la medida. Un análisis del algoritmo mixto de inferencia exacta, para el caso en el que se emplean redes bayesianas para modelar filtros de Kalman conmutados, puede encontrarse en [MurphyKP1998c]. El análisis de dicho artículo es directamente extensible al problema de la asimilación de medidas en función de una probabilidad de validación, ya que en nuestro planteamiento, este problema es modelada mediante una red bayesiana que incorpora los dos modelos de medida de la tabla C2-11, y por lo tanto, es un problema resoluble por un filtro de Kalman conmutado en el modelo de medida.

▪ ***Tests de validación basados en la distancia entre el valor estimado y real de la medida***

Entre los diferentes tests que se pueden realizar durante la etapa de validación, existe un grupo que basa su decisión sobre si la medida debe de ser aceptada o rechazada en la distancia (discrepancia) existente entre el valor estimado de la medida (calculado a partir del valor estimado del estado del sistema) y su valor real. Los tests basados en la distancia no puede ser modelados gráficamente dentro del ámbito de las redes bayesianas ya que implicarían la inclusión de algún enlace (por ejemplo la línea punteada de la figura C2-5) que: (1) modelase la relación de dependencia existente entre la variable de validación $\mathbf{v}_{s,t}$ y la distancia entre el valor estimado y real de la medida $\mathbf{z}_{s,t}$, y (2) rompiese la estructura acíclica obligatoria de la red. Además del problema gráfico, las probabilidades condicionales de las redes bayesianas mixtas con algoritmos de inferencia exacta no permiten modelar los tests de validación basados en la distancia (Euclídea, Mahalanobis, Bhattacharaya y Chernoff [Fukunaga1993]) ni que un nodo discreto tenga padres continuos.

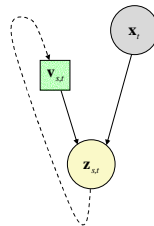


Figura C2-5. Inclusión de un enlace para modelar la relación de dependencia existente entre la variable de validación y la distancia entre el valor estimado y real de la medida.

Por último, hay que tener en cuenta que el enlace punteado de la figura C2-5 únicamente representa la existencia de una relación de dependencia entre la variable de validación $\mathbf{v}_{s,t}$, y la distancia entre el valor estimado y real de la medida $\mathbf{z}_{s,t}$, y que por lo tanto podría ser necesario incluir otros nodos y enlaces auxiliares para modelar correctamente dicha relación.

Aunque estos tests no pueden ser modelados dentro del ámbito de las redes bayesianas, pueden considerarse como un proceso ajeno a los algoritmos de inferencia exacta (C2-2 y C2-3) y una vez realizados utilizar sus resultados dentro de dichos algoritmos. Eso si, esta forma de proceder debe tener siempre en cuenta que el resultado de la validación depende del valor estimado de la medida (y por lo tanto del valor estimado del estado) y que a su vez el resultado del proceso de absorción de la medida depende del resultado obtenido durante la validación. Es decir, que usamos la variable que queremos estimar (estado del sistema \mathbf{x}_t) para decidir si la medida es válida o no, y el valor de las medidas consideradas válidas para mejorar el valor estimado del sistema.

La distancia de Mahalanobis, que pondera la separación existente entre el valor de medida estimado y real con la covarianza inversa del valor estimado de la medida, es una de las más utilizadas para el test de validación ya que es fácil de calcular (ecuación (C2-34)) y tiene una forma equivalente al test chi-cuadrado, que contrasta la veracidad de que se cumpla la hipótesis $\zeta_{s,t} = \hat{\mathbf{z}}_{s,t}$. Esta propiedad permite igualar el tamaño de la región de validación γ al estadístico $\chi^2_{p,n}$, que representa el valor inverso de la distribución chi-cuadrado para un intervalo de confianza p y una variable de n grados de libertad. Por lo tanto, la elección de este valor como umbral nos permitirá realizar un test de validación que, dado un valor estimado y real de la medida y la covarianza inversa asociado al valor estimado, será pasado por un tanto por ciento p de las medidas.

$$\zeta_{s,t} \text{ valida} \Leftrightarrow (\zeta_{s,t} - \hat{\mathbf{z}}_{s,t})^T \mathbf{S}_{s,t}^{-1} (\zeta_{s,t} - \hat{\mathbf{z}}_{s,t}) \leq \gamma, \quad \text{con} \quad \mathbf{S}_{s,t} = \mathbf{H}_t \mathbf{Y}_t^{-1} \mathbf{H}_t^T + \mathbf{R}_t \quad (\text{C2.34})$$

Antes de realizar la etapa de validación en los tests que utilizan el valor estimado de la medida, es necesario extraer dicha información de los potenciales del algoritmo de inferencia exacto que la almacenan. El hecho de que la validación en si se considere un proceso externo al algoritmo de inferencia, no implica que no se pueda utilizar el algoritmo de inferencia exacta para obtener de forma automática, antes de la etapa de validación de la medida $\zeta_{s,t}$, el valor estimado de la medida $\hat{\mathbf{z}}_{s,t}$ y covarianza $\mathbf{S}_{s,t}$. Para obtener esos valores tan sólo es necesario incluir en el algoritmo de inferencia exacta antes de la etapa de validación una absorción tipo (4) de la figura C2-4 (ya que ésta permitirá que el nodo sensorial $\mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$ absorba la información existente sobre el estado del sistema en el nodo de transición $\mathbf{x}_{t-1} \mathbf{x}_t$) y marginalizar la variable asociada al estado \mathbf{x}_t del potencial válido ($\mathbf{v}_{s,t}=\text{true}$) asociado al conglomerado sensorial $\mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$.

El resultado de estas operaciones se recoge en las siguientes expresiones, en las que se supone que la medida $\zeta_{s,t}$ (realizada por el sensor s en el instante t) llega al algoritmo después de que se haya

realizado la k -ésima etapa de inferencia. La absorción tipo (4) comienza con la operación de marginalización (C2-35) de la variable \mathbf{x}_{t-1} del potencial asociado al conglomerado de transición de estados $\mathbf{x}_{t-1}\mathbf{x}_t$. El resultado de esta marginalización se analizó en las expresiones (C2-19) y (C2-27), la primera para el caso en el que no había llegado ninguna medida al algoritmo y la segunda cuando el algoritmo disponía de toda la información sensorial. En el caso que ahora nos ocupa, supondremos que las medidas se pueden realizar de forma asíncrona y temporalmente desordenada, y que antes de la llegada al algoritmo de la medida $\zeta_{s,t}$ no ha llegado a éste ninguna medida de otro sensor correspondiente al mismo instante de muestro t . Por lo tanto, en $\hat{\mathbf{y}}_t(k)$ e $\mathbf{Y}_t(k)$ se recoge el valor estimado y la covarianza de la información del estado del sistema para el instante t tras realizar una etapa equivalente a la predicción del filtro de Información sobre el valor estimado $\hat{\mathbf{y}}_{t-1}(k)$ y la covarianza $\mathbf{Y}_{t-1}(k)$ de la información del estado del sistema para el instante $t-1$, valores donde se almacenan toda la información sensorial \mathbf{Z} medida antes del instante t y recibida por el algoritmo hasta su k -ésima etapa. La siguiente operación de la etapa de absorción, ecuación (C2-36), consiste en la actualización del potencial del separador $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$ con el valor obtenido tras la marginalización. Por último, la operación (C2-37) es la actualización del potencial del conglomerado $\mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$ asociado a la medida realizada, cuando se considera que ésta es válida ($\mathbf{v}_{s,t} = \text{true}$). La misma expresión se puede utilizar para el caso no válido ($\mathbf{v}_{s,t} = \text{false}$) sustituyendo las componentes en las que aparece la matriz $\mathbf{H}_{s,t}$ por submatrices iguales a $\mathbf{0}$. En esta actualización, se le añade el nuevo valor del potencial del separador $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$ y se le quita el valor anterior, que es igual al de la etapa de inicialización ya que hasta el momento en el que la medida $\zeta_{s,t}$ llega al algoritmo de inferencia no se han realizado absorciones a través de dicho separador.

$$\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}^{k*}(\mathbf{x}_t, \mathbf{Z}) = \int \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}^k(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{Z}) d\mathbf{x}_{t-1} = (\mathcal{G}, [\hat{\mathbf{y}}_t(k)], [\mathbf{Y}_t(k)]) \quad (\text{C2.35})$$

$$\Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}^{k+1}(\mathbf{x}_t, \mathbf{Z}) = \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}^{k*}(\mathbf{x}_t, \mathbf{Z}) = (\mathcal{G}, [\hat{\mathbf{y}}_t(k)], [\mathbf{Y}_t(k)]) \quad (\text{C2.36})$$

$$\begin{aligned} \Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^{k+1}(\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t} = \text{true}, \mathbf{Z}) &= \Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}^0(\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t} = \text{true}) \cdot \frac{\Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}^{k+1}(\mathbf{x}_t, \mathbf{Z})}{\Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}^0(\mathbf{x}_t)} \\ &= \left(\mathcal{G}, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & -\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right) \cdot \frac{(\mathcal{G}, [\hat{\mathbf{y}}_t(k)], [\mathbf{Y}_t(k)])}{(\mathcal{O}, [\mathbf{0}], [\mathbf{0}])} = \\ &= \left(\mathcal{G}, \begin{bmatrix} \hat{\mathbf{y}}_t(k) \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{Y}_t(k) + \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & -\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right) \end{aligned} \quad (\text{C2.37})$$

$$\text{con } \mathbf{Z} = \{\zeta_{I_S, J_T} \mid I_S \in \{1, \dots, S\}, J_T \in \{1, \dots, t-1\}\}, \hat{\mathbf{y}}_0(k) = \hat{\mathbf{y}}_0 \quad e \quad \mathbf{Y}_0(k) = \mathbf{Y}_0$$

Después del proceso de absorción, el valor estimado de la medida se puede obtener marginalizando la variable de estado \mathbf{x}_t del potencial válido ($\mathbf{v}_{s,t} = \text{true}$) del conglomerado de medida $\mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$. El resultado de esta operación, que se recoge en la expresión (C2-38), es la información del valor estimado de la medida $\mathbf{S}_t^{-1}(k+1)\hat{\mathbf{z}}_{s,t}(k+1)$ y su covarianza inversa $\mathbf{S}_t^{-1}(k+1)$.

$$\Phi_{\mathbf{x}_t, \mathbf{z}_{s,t}}^{k+1*}(\mathbf{z}_{s,t}, \mathbf{v}_{s,t} = \text{true}, \mathbf{Z}) = \int \Phi_{\mathbf{x}_t, \mathbf{z}_{s,t}}^{k+1}(\mathbf{x}_t, \mathbf{z}_{s,t}, \mathbf{v}_{s,t} = \text{true}, \mathbf{Z}) d\mathbf{x}_t = \quad (\text{C2.38})$$

$$\begin{aligned} &= (\mathbf{g}, \mathbf{h}^*, \mathbf{K}^*) = (\mathbf{g}, [\mathbf{S}_t^{-1}(k+1)\hat{\mathbf{z}}_{s,t}(k+1)], [\mathbf{S}_t^{-1}(k+1)]) \\ \mathbf{K}^* &= \mathbf{R}_{s,t}^{-1} - \mathbf{R}_{s,t}^{-1}\mathbf{H}_{s,t} \left(\mathbf{Y}_t(k) + \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} \right)^{-1} \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} = \left(\mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(k) \mathbf{H}_{s,t}^T + \mathbf{R}_{s,t} \right)^{-1} = \mathbf{S}_t^{-1}(k+1) \\ \mathbf{h}^* &= \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} \left(\mathbf{Y}_t(k) + \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} \right)^{-1} \hat{\mathbf{y}}_t(k) = \\ &= \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} \left(\mathbf{Y}_t^{-1}(k) - \mathbf{Y}_t^{-1}(k) \mathbf{H}_{s,t}^T \left(\mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(k) \mathbf{H}_{s,t}^T + \mathbf{R}_{s,t} \right)^{-1} \mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(1) \right) \hat{\mathbf{y}}_t(k) = \\ &= \mathbf{R}_{s,t}^{-1} \left(\mathbf{1} - \mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(k) \mathbf{H}_{s,t}^T \left(\mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(k) \mathbf{H}_{s,t}^T + \mathbf{R}_{s,t} \right)^{-1} \right) \mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(k) \hat{\mathbf{y}}_t(k) = \\ &= \mathbf{R}_{s,t}^{-1} \left(\mathbf{1} - \left(\mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(k) \mathbf{H}_{s,t}^T + \mathbf{R}_{s,t} - \mathbf{R}_{s,t} \right) \left(\mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(k) \mathbf{H}_{s,t}^T + \mathbf{R}_{s,t} \right)^{-1} \right) \mathbf{H}_{s,t} \hat{\mathbf{x}}_t(k) = \\ &= \mathbf{R}_{s,t}^{-1} \left(\mathbf{1} - \mathbf{1} + \mathbf{R}_{s,t} \left(\mathbf{H}_{s,t} \mathbf{Y}_t^{-1}(k) \mathbf{H}_{s,t}^T + \mathbf{R}_{s,t} \right)^{-1} \right) \hat{\mathbf{z}}_{s,t}(k+1) = \mathbf{R}_{s,t}^{-1} \mathbf{R}_{s,t} \mathbf{S}_t^{-1}(k+1) \hat{\mathbf{z}}_{s,t}(k+1) = \mathbf{S}_t^{-1}(k+1) \hat{\mathbf{z}}_{s,t}(1) \\ \text{con } \mathbf{Z} &= \left\{ \zeta_{I_S, J_T} \mid I_S \in \{1, \dots, S\}, J_T \in \{1, \dots, t-1\} \right\}, \hat{\mathbf{y}}_0(k) = \hat{\mathbf{y}}_0 \quad e \quad \mathbf{Y}_0(k) = \mathbf{Y}_0 \end{aligned}$$

Después de la marginalización de la ecuación (C2-38), se puede realizar el test de validación de la expresión (C2-34) de forma casi inmediata, tras calcular el valor estimado de la medida multiplicando la covarianza de la medida estimada $\mathbf{S}_t(k+1)$ por el valor estimado de la información de la medida $\mathbf{S}_t^{-1}(k+1)\hat{\mathbf{z}}_{s,t}(k+1)$.

Si la medida $\zeta_{s,t}$ es válida, ésta tiene que ser incluida como evidencia en el potencial del conglomerado válido de medida, $\mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$ con $\mathbf{v}_{s,t} = \text{true}$, y la nueva información sensorial absorbida por el potencial de transición de estados $\mathbf{x}_{t-1} \mathbf{x}_t$. Debido a los cambios introducidos en los potenciales del separador $\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$ y del conglomerado $\mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$ por la realización de la absorción tipo (4), los resultados de las operaciones de incorporación de la evidencia y de la absorción tipo (3) sobre estos dos potenciales no producen los mismos resultados que en el caso en el que la absorción tipo (4) no se realizaba (análisis realizado en la sección de inferencia exacta y fusión sensorial asíncrona y/o temporalmente retardada sobre la red bayesiana que modela un sistema lineal de control con múltiples sensores).

El resultado de la incorporación de la medida $\zeta_{s,t}$ como evidencia en el potencial del conglomerado válido de medida, $\mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$ con $\mathbf{v}_{s,t} = \text{true}$, puede verse en la expresión (C2-39), cuyo punto de partida es el valor del potencial de la expresión (C2-37).

$$\begin{aligned} \Phi_{\mathbf{x}_t, \mathbf{z}_{s,t}}^{k+1}(\mathbf{x}_t, \mathbf{v}_{s,t} = \text{true}, \mathbf{Z} \cup \zeta_{s,t}) &= \Phi_{\mathbf{x}_t, \mathbf{z}_{s,t}}^{k+1}(\mathbf{x}_t, \mathbf{z}_{s,t} \mid \mathbf{z}_{s,t} = \zeta_{s,t}, \mathbf{v}_{s,t} = \text{true}, \mathbf{Z}) = \\ &= (\mathbf{g}, [\hat{\mathbf{y}}_t(k) + \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \zeta_{s,t}], [\mathbf{Y}_t(k) + \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t}]) \\ \text{con } \mathbf{Z} &= \left\{ \zeta_{I_S, J_T} \mid I_S \in \{1, \dots, S\}, J_T \in \{1, \dots, t-1\} \right\}, \hat{\mathbf{y}}_0(k) = \hat{\mathbf{y}}_0 \quad e \quad \mathbf{Y}_0(k) = \mathbf{Y}_0 \end{aligned} \quad (\text{C2.39})$$

A continuación analizamos el resultado de la absorción tipo (3). Primero se marginaliza la variable de medida $\mathbf{z}_{s,t}$ del potencial válido ($\mathbf{v}_{s,t} = \text{true}$) del conglomerado de medida $\mathbf{x}_t \mathbf{z}_{s,t} \mathbf{v}_{s,t}$, y como éste ya no depende de dicha variable el resultado de la marginalización es el mismo que el recogido en la

expresión (C2-39). A continuación, se actualiza el valor del potencial del separador $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{z}_{s,t}\mathbf{v}_{s,t}$ con el resultado de la marginalización. Y por último, se actualiza el valor del potencial de transición de estados $\mathbf{x}_{t-1}\mathbf{x}_t$, multiplicándolo por $(g, [\hat{\mathbf{y}}_t(k) + \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \boldsymbol{\zeta}_{s,t}], [\mathbf{Y}_t(k) + \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t}])$ y dividiéndolo entre $(g, [\hat{\mathbf{y}}_t(k)], [\mathbf{Y}_t(k)])$, que son los valores nuevo y viejo del potencial del separador $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{z}_{s,t}\mathbf{v}_{s,t}$ respectivamente, de forma que el resultado de la operación es únicamente añadir la información de la medida $\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \boldsymbol{\zeta}_{s,t}$ y su covarianza $\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t}$ en el potencial de transición de estados $\mathbf{x}_{t-1}\mathbf{x}_t$. Es decir, aunque los potenciales del conglomerado de medida y del separador tienen un valor diferente que en el caso en el que no se realizaba la absorción tipo (4), el resultado del potencial de transición de estados se mantiene, hecho de esperar ya que el algoritmo de inferencia exacto utilizado es un método consistente.

En el algoritmo C2-4 se presenta una nueva versión del algoritmo de inferencia C2-3, que incluye las nuevas etapas necesarias para (1) estimar el valor de la medida y su covarianza a partir del valor estimado del estado del sistema, y (2) realizar alguno de los tests de validación basados en el valor estimado de la medida, como por ejemplo el test de validación chi-cuadrado (C2-34). La obtención del valor estimado de medida se realiza en las etapas 2.b, 2.c y 2.d, mientras que el test se realiza en la etapa 2.e, que como se considera que es ajena al algoritmo de inferencia exacta sobre la red se ha representado en letra cursiva.

- 1) Inicialización: realización de etapa 1) del algoritmo C2-1
- 2) Bucle principal: **while(true)**
 - a. Esperar hasta que haya una medida disponible $\boldsymbol{\zeta}_{s,t}$ para cualquier \mathbf{t} y \mathbf{s}
 - b. Realizar una absorción tipo (4) desde el conglomerado sensorial sobre el que se ha medido sobre su nodo de transición de estados (absorción $\mathbf{x}_{t-1}\mathbf{x}_t \rightarrow \mathbf{x}_t\mathbf{z}_{s,t}$)
 - c. Marginalizar la variable del estado \mathbf{x}_t del potencial asociado al conglomerado de medida para obtener el potencial $(\mathbf{g}, \mathbf{h}, \mathbf{K})$
 - d. Obtener el valor estimado de la medida y su covarianza como $\mathbf{K}^{-1}\mathbf{h}$ y \mathbf{K}^{-1} respectivamente
 - e. *Realizar el test de validación externa basado en la estima de la medida y su covarianza (por ejemplo el test chi-cuadrado de la expresión (C2-34))*
 - f. Si la medida es válida: realizar las etapas 2)c.i., 2)c.ii. y 2)c.iii. del algoritmo C2-3

Algoritmo C2-4. Algoritmo de inferencia exacto para la red bayesiana de la tabla C2-6 en el caso en el que las medidas puedan ser no válidas, asíncronas, retardadas y estar temporalmente desordenadas y se incluya mediante una instrucción condicional una etapa de validación externa basada en la estimación de la medida (como por ejemplo el test chi-cuadrado).

- **Tests de validación basados en la distancia entre el valor estimado y real de la medida para un sistema con medidas asíncronas y temporalmente desordenadas.**

En la última demostración se realizó la suposición de que antes de la llegada al algoritmo de inferencia de la medida $\zeta_{s,t}$, realizada por el sensor s en el instante de muestreo t , éste no había recibido ninguna medida $\zeta_{k,t}$ de otro sensor k realizada en el mismo instante t . Esta restricción se impuso para asegurar que la información de la variable del estado $\hat{\mathbf{y}}_t(k)$ y su covarianza $\mathbf{Y}_t(k)$ recogían únicamente el valor estimado tras una etapa equivalente a la predicción del filtro de Información sobre la información de la variable de estado y su covarianza, $\hat{\mathbf{y}}_{t-1}(k)$ e $\mathbf{Y}_{t-1}(k)$, para el instante de muestreo anterior $t-1$, y nunca información sensorial ($\hat{\mathbf{i}}_{j,t}$ e $\hat{\mathbf{i}}_{j,t}$) del instante de muestreo t de cualquier otro sensor distinto de s ($k \neq s$). Por lo tanto, la restricción asegura que la información sensorial de un sensor k para el instante t no es utilizada para validar la información sensorial de otro sensor s en el mismo instante t , evitándose de este modo que los sensores que son capaces de procesar y comunicar su información sensorial al algoritmo de inferencia con mayor celeridad tengan preferencia a la hora de validar sus medidas dentro del mismo instante de muestreo.

Sin embargo, esta restricción no puede ser sostenida por un algoritmo de inferencia para un sistema con medidas asíncronas y temporalmente desordenadas que tiene como objetivo devolver en todo momento el valor estimado más completo del estado del sistema, entendiéndose como valor estimado completo aquel que contiene toda la información sensorial recibida por el algoritmo de inferencia. La solución más sencilla consiste en eliminar la restricción, tal y como ocurre en el algoritmo C2-4, teniendo siempre en cuenta que la falta de la misma implica que no todos los sensores tienen la misma preferencia a la hora de que sus medidas sean validadas por el algoritmo dentro del mismo instante de muestreo. Otra forma de solucionar el problema consistiría en buscar una red bayesiana algo más compleja, con algunos nodos auxiliares, que permitiese modelar directamente el comportamiento deseado del algoritmo.

A medio camino entre las dos soluciones, existe una que consiste en quitar la restricción y eliminar temporalmente del potencial asociado al conglomerado de transición de estados $\mathbf{x}_{t-1}\mathbf{x}_t$ la información sensorial ($\hat{\mathbf{i}}_{k,t}$ e $\hat{\mathbf{i}}_{k,t}$) procedente del resto de los sensores k antes de realizar la absorción tipo 4. La eliminación de la información sensorial de otros sensores es una labor sencilla, que se puede obtener realizando una operación adicional y la absorción tipo 4 sobre un conjunto de variables auxiliares. Antes de exponer la operación adicional, es importante recordar que en el algoritmo de inferencia C2-1, en el que no se realizaban absorciones tipo (4), los potenciales de los separadores de medida $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{z}_{k,t}\mathbf{x}_t$ almacenaban antes de realizar una medida los valores de la inicialización $(0, [0], [0])$, y, después de medir, los valores de la información sensorial

$(g, [i_{j,t}], [I_{j,t}])$. Por lo tanto, si no se realizan absorciones tipo (4) sobre los conglomerados y separadores del algoritmo es posible eliminar temporalmente la información sensorial del instante de muestreo del conglomerado de transición de estados $\mathbf{x}_{t-1}\mathbf{x}_t$ dividiendo el valor de su potencial entre el valor de todos los separadores de medida $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{z}_{k,t}\mathbf{x}_t$. El resultado será almacenado sobre un potencial auxiliar $\Phi_{T_AUX}(\mathbf{x}_{t-1}, \mathbf{x}_t)$, sobre el que se realizará una absorción tipo (4) que utilice los potenciales del árbol de unión sin actualizar su valor. El resultado de la transición se almacenará en la variable auxiliar $\Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$, sobre la que se realizará la marginalización que permite estimar la información del valor de la medida y su covarianza.

De esta forma se asegura que si dos medidas $\zeta_{s,t}$ y $\zeta_{k,t}$ son realizadas por dos sensores diferentes en el mismo instante de tiempo, y una de ellas, por ejemplo $\zeta_{k,t}$, es procesada por el algoritmo antes que la otra $\zeta_{s,t}$ (comportamiento esquematizado en la figura siguiente), el resultado de procesar la primera cuando sea válida no va a influir en la validación de la procesada posteriormente.

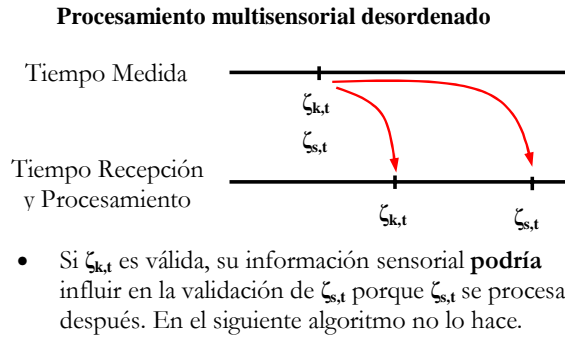


Figura C2-6. La validación temporalmente desordenada de medidas tomadas por distintos sensores en el mismo instante de tiempo podría influir en la validación de las medidas procesadas más tarde.

El algoritmo C2-5 recoge las modificaciones propuestas, que evitan que las medidas tomadas en el mismo instante de muestreo influyan en la validación de las medidas tomadas en dicho instante que se procesan más tarde que el resto:

- 1) Inicialización
 - a. Obtener el modelo del sistema (ecuación de transición de estados) y de los sensores (ecuaciones sensoriales).
 - b. Construir la red bayesiana, con tantos nodos sensoriales como sean necesarios y su árbol de unión correspondiente.
 - c. Convertir las ecuaciones probabilísticas de transición de estado y sensoriales en potenciales condicionales.
 - d. Inicializar los potenciales de los conglomerados (a partir de los potenciales condicionales) y de los separadores (a 0).
 - e. Realizar absorciones tipo (1) de forma sucesiva y ordenada de izquierda a derecha desde el primer nodo de transición de estados hasta el último (primera absorción $\mathbf{x}_0\mathbf{x}_1 \rightarrow \mathbf{x}_1\mathbf{x}_2$, siguiente absorción $\mathbf{x}_1\mathbf{x}_2 \rightarrow \mathbf{x}_2\mathbf{x}_3$, ..., y última absorción $\mathbf{x}_{n-2}\mathbf{x}_{n-1} \rightarrow \mathbf{x}_{n-1}\mathbf{x}_n$)

- 2) Bucle principal: **while(true)**
- Esperar hasta que haya una medida disponible $\zeta_{s,t}$ para cualquier t y s
 - Eliminar la información sensorial del potencial de transición de estados
- $$\Phi_{T_AUX}(\mathbf{x}_{t-J}, \mathbf{x}_t) = \frac{\Phi_{\mathbf{x}_{t-J}\mathbf{x}_t}(\mathbf{x}_{t-J}, \mathbf{x}_t)}{\prod_s \Phi_{\mathbf{x}_{t-J}\mathbf{x}_t \cap \mathbf{x}_{t-J}, \mathbf{x}_t}(\mathbf{x}_t)}$$
- Realizar una absorción tipo (4) desde el nodo sensorial sobre el que se ha medido del $\Phi_{T_AUX}(\mathbf{x}_{t-J}, \mathbf{x}_t)$, sin actualizar los valores del potencial del separador de medida ni el potencial de medida transición de estados. Almacenar el resultado en $\Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$
 - Marginalizar la variable del estado \mathbf{x}_t del potencial $\Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$. El resultado de la operación será el potencial $\Phi = (\mathbf{g}, \mathbf{h}, \mathbf{K})$
 - Obtener el valor estimado de la medida y su covarianza como $\mathbf{K}^{-1}\mathbf{h}$ y \mathbf{K}^{-1} respectivamente
 - Realizar el test de validación externa basado en la estima de la medida y su covarianza (por ejemplo el test chi-cuadrado de la expresión (C2-34))
 - Si la medida es válida
 - Introducir la medida $\zeta_{s,t}$ como evidencia en el nodo sensorial $\mathbf{x}_{s,t}\mathbf{z}_t$
 - Realizar una absorción tipo (3) desde el nodo de transición de estados asociado al nodo sensorial sobre el que se ha medido (absorción $\mathbf{x}_t\mathbf{z}_{s,t} \rightarrow \mathbf{x}_{t-1}\mathbf{x}_t$)
 - Si $t \neq n$, realizar absorciones tipo (1) de forma sucesiva y ordenada de izquierda a derecha desde el nodo de transición de estados asociado a la medida disponible hasta el último (primera absorción $\mathbf{x}_{t-1}\mathbf{x}_t \rightarrow \mathbf{x}_t\mathbf{x}_{t+1}$, ... , y última absorción $\mathbf{x}_{n-2}\mathbf{x}_{n-1} \rightarrow \mathbf{x}_{n-1}\mathbf{x}_n$)

Algoritmo C2-5. Algoritmo de inferencia exacto para la red bayesiana de la tabla C2-6 en el caso en el que las medidas puedan ser no válidas, asíncronas, retardadas y estar temporalmente desordenadas, y se incluya mediante una instrucción condicional una etapa de validación externa, basada en la estimación de la medida (como por ejemplo el test chi-cuadrado), que no utilice la información sensorial proveniente, para el mismo instante de muestreo, de otros sensores con el objeto de validar la información sensorial en dicho instante.

Esta modificación resuelve el problema de la validación temporalmente desordenada de las medidas recogidas por los distintos sensores dentro de un mismo periodo de muestreo t , pero no el problema que existe ante un posible desorden temporal en la validación entre diferentes sensores y distintos instantes de medida. Si dos medidas realizadas por dos sensores en los instantes i y j , con j mayor que i , llegan temporalmente desordenadas al algoritmo de inferencia, es decir la medida j antes de la i , la validación de la medida j no utilizará la información sensorial de la medida i al no estar ésta asimilada en el valor estimado del estado del sistema. Si se volviese a realizar el test de validación de la medida j una vez que el algoritmo hubiese asimilado la medida i , podría darse el caso que el nuevo resultado del test fuese diferente. Este comportamiento se esquematiza en la siguiente figura.

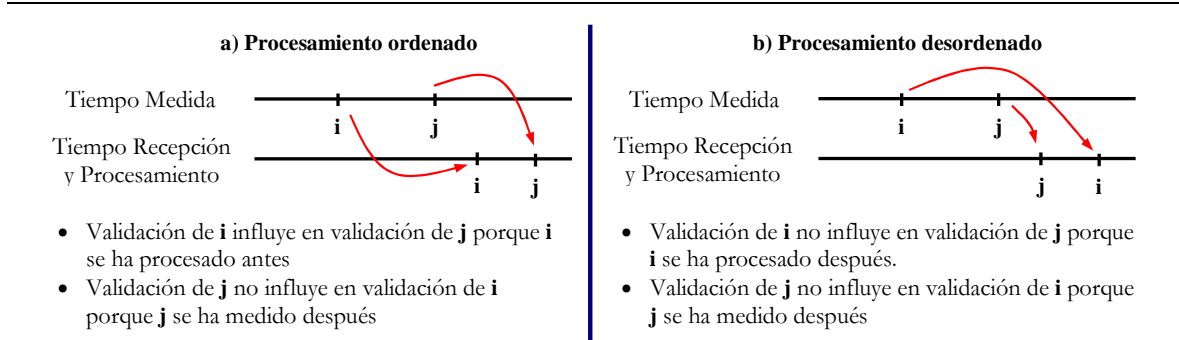


Figura C2-7. Influencia en el proceso de validación del orden de procesamiento de medidas tomadas por el mismo o distintos sensores en distintos instantes de tiempo.

Aunque es posible modificar el algoritmo de inferencia (para volver a realizar los tests de validación sobre medidas realizadas en periodos de muestreo posteriores a la nueva medida recibida por el algoritmo, y actualizar el estado del sistema con el resultado de dichos tests) esta modificación no es justificable cuando se considera que el filtro se encuentra bien sintonizado y que la utilidad del test de validación es rechazar las escasas medidas no válidas, cuyo error no ha sido modelado por las ecuaciones del modelo de medida y sus covarianza respectivas. Además, hay que tener en cuenta la sobrecarga computacional que implicaría la continua revalidación de medidas en un sistema donde el desorden temporal de asimilación se da de forma habitual. Por lo tanto, nuestro algoritmo de inferencia no modificará el estado de validación que se asocia a la medida tan pronto como ésta llega al algoritmo de inferencia, manteniéndose la solución presentada en la versión del algoritmo (C2-5).

Una modificación interesante, relativa al estudio del correcto funcionamiento del algoritmo de inferencia, sería el almacenar todas las medidas realizadas por el algoritmo de forma que se pudiese realizar una análisis posterior de la influencia de la validación de medidas cuando éstas se realizan de forma desordenada (durante el funcionamiento normal del algoritmo) y ordenada (estudio a posteriori). Ante una diferencia significativa en las medidas aceptadas y rechazadas en cada ejecución del algoritmo, habría que estudiar el motivo de la diferencia para extraer algún tipo de información que permita mejorar el modelo sensorial del sistema.

En el capítulo IV se realizará un estudio de la influencia del retardo y desorden de las medidas en la etapa de validación, y por lo tanto, en los valores estimados por el algoritmo.

Capítulo III

Redes bayesianas dinámicas para modelar sistemas de control con sensores asíncronos y medidas temporalmente retardadas

En el capítulo anterior se presentó una red bayesiana que modela el comportamiento de un sistema lineal dinámico, cuyas señales de control se conocen a priori, y cuyos sensores lineales y asíncronos proporcionan la información sensorial de forma temporalmente desordenada. Sobre dicha red, se presentaron un conjunto de algoritmos de inferencia exacta capaces de estimar el estado del sistema en un instante de muestreo a partir de toda la información sensorial tomada en instantes de muestreo anteriores. Además, se demostró la equivalencia entre dichos algoritmos y el filtro de Información para sistemas multisensoriales, cuyo funcionamiento se fundamenta en la asimilación temporalmente ordenada de la información sensorial. Debido a las restricciones impuestas durante las demostraciones, el sistema de fusión multisensorial construido sobre dicha red y los algoritmos de inferencia desarrollados sólo pueden ser aplicados sobre un número limitado de casos.

La primera restricción, el conocimiento a priori de las señales de control, se puede eludir fácilmente, sustituyendo la red bayesiana propuesta en el capítulo anterior por una red bayesiana dinámica equivalente. La nueva red se inicializará con un único nodo, el nodo de estado asignado al instante 0 , y de forma progresiva, para cada nuevo instante de muestreo t , se incluirán el nodo de estado \mathbf{x}_t asociado a dicho instante, sus nodos sensoriales $\mathbf{z}_{s,t}$ y de validación $\mathbf{v}_{s,t}$, y se inicializarán los potenciales correspondientes a los conglomerados asociados a los nodos añadidos. De esta forma, la señal de control \mathbf{u}_{t-1} podrá ser incluida cíclicamente, como parte del valor inicial del nuevo potencial de transición de estados. Al mismo tiempo, para evitar el crecimiento continuo de la memoria utilizada por el algoritmo de inferencia, será conveniente ir eliminando progresivamente los nodos asociados a instantes temporales anteriores, así como sus potenciales correspondientes. Sin embargo, es necesario asegurar que la información que llega al algoritmo de inferencia de forma temporalmente retardada puede ser utilizada para mejorar el valor estimado para el estado en todos los instantes de muestreo posteriores, por lo que será necesario mantener los nodos y potenciales correspondientes a una ventana de tiempo cuyo tamaño asegure que toda la información sensorial válida puede ser utilizada.

La segunda restricción, la linealidad en las expresiones que modelan la dinámica del sistema (ecuaciones de transición de estados) y en las relaciones entre las medidas y el estado del sistema (ecuación de medida), impuesta por la elección del tipo de algoritmo de inferencia utilizado y no por la estructura de la red bayesiana, sustenta la demostración de equivalencia entre nuestros algoritmos de inferencia y el filtro de Información. Sin embargo, esta restricción no es realmente necesaria, y se puede evitar si se sustituye el algoritmo de inferencia por otro que permita realizar inferencias sobre sistemas gaussianos no lineales.

El filtro de Información Extendido es la versión del filtro de Información para el caso no lineal, por lo que será el soporte del algoritmo de inferencia desarrollado para nuestro problema. Aún más, el uso del filtro de Información para el caso lineal cuando las medidas están temporalmente ordenadas resulta computacionalmente ventajoso frente al uso del algoritmo desarrollado en el capítulo anterior, ya que las operaciones que hay que realizar sobre el mismo son más sencillas. Pero el uso directo de los filtros de Información para un sistema asíncrono con medidas temporalmente desordenadas no es posible, ya que éstos asimilan la información sensorial cíclicamente y de forma temporalmente ordenada. Por lo tanto, se propondrán dos nuevos algoritmos, uno para el caso lineal y otro para el no lineal, que, basándose en las operaciones de los filtros de Información y en la forma en la que el algoritmo de inferencia exacto del capítulo anterior asimila la información temporalmente desordenada, permitirán estimar el estado de sistemas de control lineales o no lineales, cuya información pueda estar disponible de forma temporalmente desordenada. El desarrollo de los dos nuevos algoritmos tendrá como objetivo principal asegurar que sus resultados sean equivalentes a los obtenidos por los algoritmos clásicos de Información y de Información Extendido, cuando a éstos últimos se les proporcione la misma información sensorial que a los primeros, pero de forma temporalmente ordenada y sin retardos.

Las principales características que diferencian a los tres algoritmos propuestos en este capítulo para la asimilación de medidas retardadas y desordenadas, que se encuentran también recogidos en el apéndice II de la tesis, se muestran en la siguiente tabla. Además, se especifica la sección del capítulo en la que el algoritmo es presentado y analizado.

Algoritmo	Ecuaciones de los Modelos	Algoritmo Soporte	Sección
C3-1 (Bayes)	Lineales	Algoritmo Exacto de Inferencia basado en árboles de unión	C3.1
C3-2 (IFAsyn)	Lineales	Filtro de Información	C3.2.1
C3-3 (IFAsynMixto)	Lineales y No Lineales	Filtro de Información Extendido	C3.2.2

Tabla C3-1. Algoritmos propuestos durante este capítulo. En la primera columna se presenta el identificador del algoritmo (C3-*) junto con el nombre utilizado en los capítulos sucesivos para hacer referencia a los mismos. En la segunda, el tipo de ecuaciones de modelos que soporta. En la tercera, el algoritmo básico sobre el que se ha desarrollado ésta nueva versión que permite asimilar información retardada y desordenada. Y en la última, la sección de este capítulo en la que cada algoritmo es presentado y analizado.

Por último se comparan los filtros de Información desarrollados y presentados en este capítulo con un amplio conjunto de algoritmos presentes en la literatura que han sido desarrollados para resolver el problema de la asimilación de medidas desordenadas y temporalmente retardadas.

C3.1. Redes bayesianas dinámicas para modelar el comportamiento de un sistema lineal con sensores asíncronos con medidas temporalmente desordenadas

En todas las demostraciones del capítulo anterior se ha supuesto que se conocía a priori el número de instantes de muestreo sobre los que se iba a estimar el estado del sistema (ya que la red bayesiana elegida para modelar su comportamiento tenía $n+1$ nodos de estado) y el valor de la señal de control para todos ellos (al ser éste valor necesario durante la etapa de inicialización de los potenciales de transición de estados).

Estas suposiciones, válidas para algunos casos, no tienen sentido en aquellos en los que se está realizando un control en lazo cerrado del sistema y/o no se pueda determinar su tiempo de funcionamiento, y, por lo tanto, no se conozca a priori los valores de la señal de control ni el número de instantes de muestreo n necesarios. Además, aún en los casos en los que las suposiciones fuesen válidas, para los casos en los que el número n de periodos de muestreo necesarios sea elevado, habría que reservar una cantidad significativa de memoria para almacenar las variables de los potenciales (g, h, K) de todos y cada uno de los conglomerados y separadores de transición de estados y de medida.

Este problema se puede solucionar sustituyendo la red bayesiana del capítulo anterior por una red bayesiana dinámica equivalente, cuya estructura evoluciona con el paso del tiempo, por medio de la inclusión de nuevos nodos y enlaces, y la eliminación de algunos de los ya existentes. Los algoritmos de inferencia exactos para redes bayesianas dinámicas utilizan las operaciones sobre potenciales canónicos presentadas en el capítulo anterior, salvo para la etapa de inicialización de los potenciales asociados a los conglomerados y separadores. Esta inicialización se irá produciendo según aparezcan nuevos elementos en el árbol de unión debido a la incorporación de nuevos nodos en la red bayesiana dinámica. Es decir, el árbol de conglomerados y separadores también se construye de forma dinámica, y los potenciales de los nuevos elementos del árbol son iniciados según los nuevos elementos aparecen, de acuerdo con las distribuciones condicionales asociadas a las relaciones probabilísticas entre nodos de la red. La nomenclatura utilizada a lo largo de esta sección es la propuesta y seguida en el capítulo anterior.

Para modelar el comportamiento de nuestro sistema de control multisensorial, se le añadirá a la red dinámica, para cada periodo de muestreo t , los nodos de estado \mathbf{x}_t , medida $\mathbf{z}_{s,t}$ y validación $\mathbf{v}_{s,t}$ asociados a dicho instante, además de todos los enlaces existentes entre dichos nodos, y entre el nuevo nodo de estado \mathbf{x}_t y el nodo de estado del instante anterior \mathbf{x}_{t-1} . En la figura C3-1, se han

destacado todos los elementos que se añaden a la red para el instante t sobre la forma sombreada amarilla de la derecha.

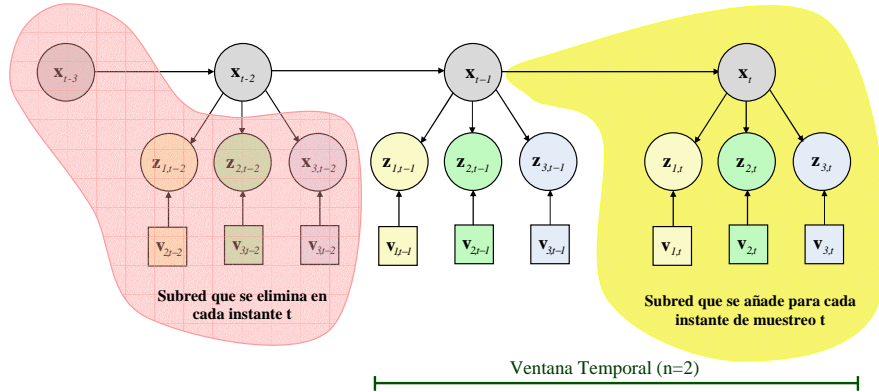


Figura C3-1. Construcción de una red bayesiana dinámica para modelar el comportamiento de un sistema de control con múltiples sensores asíncronos cuyas medidas pueden estar temporalmente desordenadas y/o ser no válidas, con una ventana de tiempo formada por dos instantes de muestreo.

Este incremento de nodos y enlaces en la red bayesiana de la figura C3-1, se traduce en la aparición, en su árbol de unión asociado, del conglomerado de transición de estados $\mathbf{x}_{t-1}\mathbf{x}_t$, los conglomerados de medida y validación $\mathbf{x}_t\mathbf{z}_{s,t}\mathbf{v}_{s,t}$, y los separadores existentes entre el conglomerado de transición de estados y los conglomerados de medida y validación $\mathbf{x}_t\mathbf{z}_{s,t}\mathbf{v}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t$. Además, hay que añadir un separador de transición de estados, pudiéndose realizar esta inserción de dos forma diferentes: (1) añadiendo el nodo de transición de estados del instante anterior $\mathbf{x}_{t-2}\mathbf{x}_{t-1} \cap \mathbf{x}_{t-1}\mathbf{x}_t$, o (2) añadiendo el nodo de transición de estados que será necesario para el próximo instante de muestreo $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}$. Aunque ambas opciones son igualmente válidas, elegiremos la segunda ya que ésta permite utilizar el separador $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}$ para extraer la información existente sobre el estado del sistema en el instante t . En la figura C3-2, se han destacado todos los elementos que se añaden al árbol de unión para el instante t sobre la forma sombreada amarilla de la derecha.

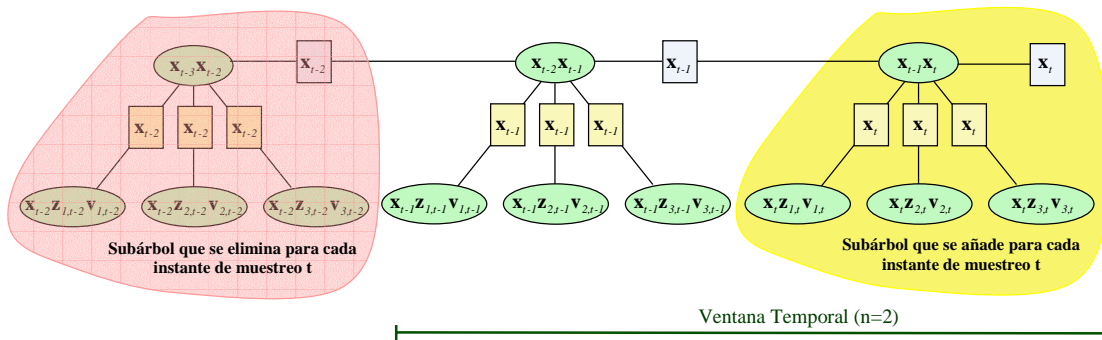


Figura C3-2. Árbol de unión asociado a la red bayesiana de la figura C3-1.

C3.1. Redes bayesianas dinámicas para modelar el comportamiento de un sistema lineal con sensores asíncronos con medidas temporalmente desordenadas

La incorporación sucesiva de nuevos nodos en la red, y por lo tanto de conglomerados y separadores en el árbol de unión, conlleva un aumento de las necesidades de memoria del algoritmo de inferencia, debido a la inclusión continuada de nuevos potenciales. Este hecho, que puede provocar el desbordamiento de la memoria asignada al algoritmo, puede ser solucionado fácilmente, eliminando, a la vez que se incluyen nuevos nodos, los nodos y enlaces asociados a instantes de muestreo lejanos y cuya desaparición no impida incluir información sensorial que llega al algoritmo de forma temporalmente retardada. El método de eliminación más sencillo consiste en:

- Elegir una ventana temporal de un tamaño n , lo suficientemente grande para asegurar que la información sensorial $\zeta_{s,k}$, que llega al algoritmo temporalmente retardada, ha sido medida en algún instante k comprendido entre el instante de muestreo actual t y el instante de muestreo $t-n+1$.
- Eliminar, a la vez que se añaden los nuevos nodos de la red (y conglomerados, separadores y potenciales de su árbol de unión), los nodos (y conglomerados, separadores y potenciales de su árbol de unión) asociados al instante de muestreo $t-n$.

En las primeras etapas del algoritmo no se realizará eliminación alguna, ya que ésta solo tiene sentido a partir del instante t que cumple que $t-n > 0$. En las figuras C3-1 y C3-2 se han destacado, sobre la forma sombreada rosada a la izquierda de las mismas, todos los elementos que hay que eliminar, antes de la incorporación de los elementos del instante t , cuando se ha elegido una ventana temporal $n=2$.

Como consecuencia del cambio de tipo de red bayesiana, hay que desarrollar nuevas versiones, para los algoritmos del capítulo anterior, que operen sobre el árbol de unión dinámico en vez de sobre el árbol de unión estático. En este capítulo únicamente se presentan las modificaciones necesarias para la última de las versiones propuestas en el capítulo anterior. Dicha versión permitía obtener el valor estimado del estado de sistemas lineales de control para todos los instantes de muestreo, cuyos sensores pueden proporcionar medidas asíncronas, no siempre válidas, y temporalmente retardadas y desordenadas. La etapa de la validación de dicho algoritmo se implementa con una instrucción condicional, de forma que sólo es necesario trabajar con la parte del potencial de medida para el que $v_{s,t} = \text{true}$. El método de validación implementado utiliza el valor estimado de la medida a través de un potencial auxiliar que evita que la información sensorial $\zeta_{s,k}$ se valide con la información sensorial proveniente de otros sensores para el mismo instante de muestreo k .

A continuación se presenta el algoritmo de inferencia exacta desarrollado para nuestro problema cuando éste se encuentra modelado mediante la red bayesiana dinámica de la figura C3-1. En este

caso se han obviado las etapas de construcción de la red bayesiana dinámica, ya que éstas no son necesarias cuando se sabe como construir dinámicamente su árbol de unión correspondiente.

1) Inicialización

- a. Obtener el modelo del sistema (ecuación de transición de estados) y de los sensores (ecuaciones de medida).
- b. Fijar el tamaño de la ventana de tiempo n .
- c. $t=1$
- d. Construir el árbol de unión inicial, formado únicamente por el subárbol asociado al instante de muestro $t=1$ (es decir, por $\mathbf{x}_0\mathbf{x}_1$, $\mathbf{x}_1\mathbf{z}_{s,1}$, $\mathbf{x}_1\mathbf{z}_{s,1}\cap\mathbf{x}_0\mathbf{x}_1$ y $\mathbf{x}_0\mathbf{x}_1\cap\mathbf{x}_1\mathbf{x}_2$).
- e. Inicializar el valor de los potenciales de los nodos del árbol:

$$\Phi_{\mathbf{x}_0\mathbf{x}_1}(\mathbf{x}_0, \mathbf{x}_1) = \Phi(\mathbf{x}_0) \cdot \Phi(\mathbf{x}_0, \mathbf{x}_1) = \left(g, \begin{bmatrix} \hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \\ \mathbf{Q}_0^{-1} \mathbf{u}_0 \end{bmatrix}, \begin{bmatrix} \hat{\mathbf{Y}}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 & -\mathbf{F}_0^T \mathbf{Q}_0^{-1} \\ -\mathbf{Q}_0^{-1} \mathbf{F}_0 & \mathbf{Q}_0^{-1} \end{bmatrix} \right)$$

$$\Phi_{\mathbf{x}_1\mathbf{z}_{s,1}}(\mathbf{x}_1, \mathbf{z}_{s,1}) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,1}^T \mathbf{R}_{s,1}^{-1} \mathbf{H}_{s,1} & -\mathbf{H}_{s,1}^T \mathbf{R}_{s,1}^{-1} \\ -\mathbf{R}_{s,1}^{-1} \mathbf{H}_{s,1} & \mathbf{R}_{s,1}^{-1} \end{bmatrix} \right)$$

$$\Phi_{\mathbf{x}_1\mathbf{z}_{s,1}\cap\mathbf{x}_0\mathbf{x}_1}(\mathbf{x}_1) = (g, [\mathbf{0}], [\mathbf{0}]) \quad \Phi_{\mathbf{x}_0\mathbf{x}_1\cap\mathbf{x}_1\mathbf{x}_2}(\mathbf{x}_1) = (g, [\mathbf{0}], [\mathbf{0}])$$

- f. Actualizar el valor del potencial del separador $\mathbf{x}_0\mathbf{x}_1\cap\mathbf{x}_1\mathbf{x}_2$ a partir de una marginalización sobre el potencial del conglomerado $\mathbf{x}_0\mathbf{x}_1$

$$\Phi_{\mathbf{x}_0\mathbf{x}_1\cap\mathbf{x}_1\mathbf{x}_2}(\mathbf{x}_1) = \int \Phi_{\mathbf{x}_0\mathbf{x}_1}(\mathbf{x}_0, \mathbf{x}_1) d\mathbf{x}_0$$

2) Bucle principal: **while**(true)

- a. Si ha pasado el tiempo de un muestreo, hay que añadir un subárbol nuevo, y posiblemente eliminar las variables del subárbol más antiguo:
 - i. $t=t+1$
 - ii. Añadir los conglomerados y separadores asociados a t (es decir, $\mathbf{x}_{t-1}\mathbf{x}_t$, $\mathbf{x}_t\mathbf{z}_{s,t}$, $\mathbf{x}_t\mathbf{z}_{s,t}\cap\mathbf{x}_{t-1}\mathbf{x}_t$ y $\mathbf{x}_{t-1}\mathbf{x}_t\cap\mathbf{x}_t\mathbf{x}_{t+1}$)
 - iii. Inicializar el valor de los potenciales de los nuevos nodos:

$$\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = \Phi(\mathbf{x}_{t-1}, \mathbf{x}_t) = \left(g, \begin{bmatrix} -\mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \mathbf{u}_{t-1} \\ \mathbf{Q}_{t-1}^{-1} \mathbf{u}_{t-1} \end{bmatrix}, \begin{bmatrix} \mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \mathbf{F}_{t-1} & -\mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \\ -\mathbf{Q}_{t-1}^{-1} \mathbf{F}_{t-1} & \mathbf{Q}_{t-1}^{-1} \end{bmatrix} \right)$$

$$\Phi_{\mathbf{x}_t\mathbf{z}_{s,t}}(\mathbf{x}_t, \mathbf{z}_{s,t}) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & -\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right)$$

$$\Phi_{\mathbf{x}_t\mathbf{z}_{s,t}\cap\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_t) = (g, [\mathbf{0}], [\mathbf{0}]) \quad \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t\cap\mathbf{x}_t\mathbf{x}_{t+1}}(\mathbf{x}_t) = (g, [\mathbf{0}], [\mathbf{0}])$$

- iv. Incluir en el potencial de transición de estados $\mathbf{x}_{t-1}\mathbf{x}_t$ la información proveniente de instantes interiores:

$$\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) \cdot \Phi_{\mathbf{x}_{t-2}\mathbf{x}_{t-1}\cap\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1})$$

- v. Actualizar el valor del potencial del separador $\mathbf{x}_{t-1}\mathbf{x}_t\cap\mathbf{x}_t\mathbf{x}_{t+1}$ a partir de la marginalización de \mathbf{x}_{t-1} sobre el potencial del conglomerado $\mathbf{x}_{t-1}\mathbf{x}_t$:

$$\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t\cap\mathbf{x}_t\mathbf{x}_{t+1}}(\mathbf{x}_t) = \int \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) d\mathbf{x}_{t-1}$$

C3.1. Redes bayesianas dinámicas para modelar el comportamiento de un sistema lineal con sensores asíncronos con medidas temporalmente desordenadas

vi. Si $t-n>0$ hay que:

1. eliminar los elementos asociados al instante $t-n$, es decir los nodos $\mathbf{x}_{t-n-1}\mathbf{x}_{t-n}$, $\mathbf{x}_{t-n}\mathbf{z}_{s,t-n}$, $\mathbf{x}_{t-n}\mathbf{z}_{s,t-n}\cap\mathbf{x}_{t-n-1}\mathbf{x}_{t-n}$ y $\mathbf{x}_{t-n-1}\mathbf{x}_{t-n}\cap\mathbf{x}_{t-n}\mathbf{x}_{t-n+1}$, y sus potenciales asociados,
2. o reservar la memoria asociada a los mismos para el próximo periodo de muestreo $t+1$.

b. Si hay una medida $\mathbf{z}_{s,k}$ disponible, hay que comprobar si ésta se ha realizado en un k dentro de la ventana temporal elegida.

i. Si no ($k>t-n$) la medida no se puede asimilar, y es conveniente mandar un mensaje al usuario para que se detecte que con el n elegido se pierden medidas.

ii. Si ($k>t-n$) la medida se asimilará, siempre que sea válida:

1. Obtener el valor del potencial de transición auxiliar

$$\Phi_{T_AUX}(\mathbf{x}_{k-I}, \mathbf{x}_k) = \frac{\Phi_{\mathbf{x}_{k-I}\mathbf{x}_k}(\mathbf{x}_{k-I}, \mathbf{x}_k)}{\prod_s \Phi_{\mathbf{x}_k\mathbf{z}_{s,k} \cap \mathbf{x}_{k-I}\mathbf{x}_k}(\mathbf{x}_k)}$$

2. Realizar una absorción tipo (4) desde el nodo sensorial sobre el que se ha medido del $\Phi_{T_AUX}(\mathbf{x}_{k-I}, \mathbf{x}_k)$, sin actualizar los valores del potencial del separador de medida ni el potencial de medida transición de estados. Almacenar el resultado en $\Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$

3. Marginalizar la variable del estado \mathbf{x}_t del potencial $\Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$. El resultado de la operación será el potencial $\Phi = (\mathbf{g}, \mathbf{h}, \mathbf{K})$

4. Obtener el valor estimado de la medida y su covarianza como $\mathbf{K}^{-1}\mathbf{h}$ y \mathbf{K}^{-1} respectivamente

5. Realizar el test de validación externa basado en la estima de la medida y su covarianza (por ejemplo el test chi-cuadrado del capítulo anterior)

6. Si la medida es válida:

a. Introducir la medida $\mathbf{z}_{k,t}$ como evidencia en el nodo sensorial $\mathbf{x}_k\mathbf{z}_{s,k}$

b. Realizar una absorción tipo (3) desde el nodo de transición de estados asociado al nodo sensorial sobre el que se ha medido (absorción $\mathbf{x}_k\mathbf{z}_{s,k} \rightarrow \mathbf{x}_{k-1}\mathbf{x}_k$)

c. Si $k \neq t$, realizar absorciones tipo (1) de forma sucesiva y ordenada de izquierda a derecha desde el nodo de transición de estados asociado a la medida disponible hasta el último (primera absorción $\mathbf{x}_{k-1}\mathbf{x}_k \rightarrow \mathbf{x}_k\mathbf{x}_{t+1}$, ... , y última absorción $\mathbf{x}_{t-2}\mathbf{x}_{t-1} \rightarrow \mathbf{x}_{t-1}\mathbf{x}_t$)

d. Actualizar el valor del potencial del separador $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}$ a partir de una marginalización sobre el potencial del conglomerado $\mathbf{x}_{t-1}\mathbf{x}_t$

$$\Phi_{\mathbf{x}_{t-I}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}(\mathbf{x}_t) = \int \Phi_{\mathbf{x}_{t-I}\mathbf{x}_t}(\mathbf{x}_{t-I}, \mathbf{x}_t) d\mathbf{x}_{t-I}$$

Algoritmo C3-1. Algoritmo de inferencia para la red dinámica bayesiana que modela el comportamiento de un sistema de control cuyos sensores proporcionan medidas que puedan ser no válidas, asíncronas, retardadas y estar temporalmente desordenadas. La etapa de validación se incluye como una instrucción condicional y para realizarla, se estima el valor de la medida sin tener en cuenta las medidas de los otros sensores para el mismo instante de muestreo.

Podría parecer que la complejidad del algoritmo ha aumentado de forma sustancial, pero no es realmente así, ya que la diferencia real entre la versión anterior (algoritmo C2-5) y la nueva (algoritmo C3-1) reside en qué etapas del algoritmo se realiza la inicialización de los potenciales:

- Para el algoritmo C2-5 del capítulo anterior, en las etapas 1.c, 1.d y 1.e se convertían las funciones probabilísticas en potenciales, los potenciales de toda la red eran inicializados, y se realizaba una primera absorción tipo (1) a lo largo de todos los conglomerados de transición de estados, de forma sucesiva y ordenada, desde el conglomerado $\mathbf{x}_0\mathbf{x}_1$ hasta el conglomerado $\mathbf{x}_{t-1}\mathbf{x}_t$.
- Para el nuevo algoritmo C3-1, las operaciones de inicialización se distribuyen durante la etapa de inicialización y el bucle principal del algoritmo: en la etapa 1.e y 2.a.ii se convierten las funciones de probabilísticas en potenciales y se inicializan los potenciales asociados a un instante de muestreo t , y mediante las etapas 1.f, 2.a.v y 2.a.vi se realizan las primeras absorciones tipo (1) a lo largo de toda la red. El posponer parte de las operaciones para que sean realizadas en el bucle principal del algoritmo sobrecarga computacionalmente dicho bucle, pero a cambio, permite realizar un control del sistema en lazo cerrado (ya que las señales de control se introducirán cíclicamente en la etapa 2.a.ii) y ahorrar memoria cuando se necesite ejecutar el algoritmo durante muchos periodos de muestreo (porque en la etapa 2.a.vi la memoria es devuelta o reservada para el subárbol del próximo periodo de muestreo).

Al igual que las versiones del capítulo anterior del algoritmo de inferencia exacta sobre redes bayesianas, los resultados obtenidos por el algoritmo C3-1 son equivalentes a los que obtendría el filtro de Información para sistemas de control lineales multisensoriales, cuando las medidas realizadas por los distintos sensores son proporcionadas al mismo de forma temporalmente ordenada. Debido a la existencia de la etapa de validación, los resultados pueden no ser idénticos, ya que al introducir la información de forma ordenada se pueden rechazar unas medidas mientras que al introducirla de forma desordenada se pueden rechazar otras. Sin embargo, si el filtro está bien sintonizado, las discrepancias existentes entre los resultados del algoritmo de inferencia exacta C3-1 y los del filtro de Información para sistemas de control lineales multisensoriales serán despreciables.

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

El algoritmo de inferencia exacto C3-1 únicamente puede ser utilizado para sistemas de control multisensoriales con ecuaciones de transferencia de estado y de medida lineales. Esta restricción no se debe a la estructura de la red bayesiana dinámica que modela las relaciones existentes entre las distintas variables del sistema, sino que ha sido introducida en el capítulo anterior a la hora de elegir el algoritmo de inferencia exacta. Se seleccionó uno basado en el árbol de unión ya que este tipo de algoritmos permite: (1) la incorporación de la evidencia en cualquier orden, propiedad fundamental a la hora de asimilar medidas que pueden estar disponibles de forma temporalmente desordenada, y (2) inferir los valores de todas las variables de la red en cualquier etapa del algoritmo.

Esta segunda propiedad no es realmente aprovechada por nuestros algoritmos, porque únicamente inferimos los valores de las variables de estado y, en los casos en los que exista una etapa de validación cuyos resultados dependan del valor estimado de la medida, las variables de medida. Además, los algoritmos de inferencia presentados no utilizan toda la información sensorial disponible para estimar el estado del sistema en todos los instantes de muestreo t , si no sólo la información sensorial disponible que ha sido adquirida en instantes de muestreo anteriores a t , al haber eliminado las absorciones tipo (2), de derecha a izquierda, entre conglomerados de transición de estados.

Cuando las medidas son recibidas por el algoritmo de inferencia dentro de su periodo de muestreo, es decir, de forma temporalmente ordenada y sin retardos, la ventana temporal del algoritmo C3-1 puede ser $n=1$. En este caso, los resultados de dicho algoritmo y del filtro de Información son idénticos. Sin embargo, los cálculos del filtro de Información se realizan directamente sobre las variables que se desea estimar, mientras que los cálculos del algoritmo de inferencia exacto C3-1 se realizan sobre potenciales. Una comparación entre la complejidad operacional de los dos algoritmos puede encontrarse en la tabla C3-2. Desde el punto de vista del uso de memoria el filtro de Información únicamente necesita almacenar la información del estado y de su varianza, mientras que el algoritmo exacto necesita almacenar los potenciales de todos los conglomerados y separadores. En la tabla C3-3 se puede observar una comparativa de la memoria utilizada por los dos algoritmos. Por lo tanto, la complejidad de los cálculos realizados por el algoritmo de inferencia C3-1 cuando no hay retardos en las medidas es mayor, así como la memoria necesaria para llevarlos a cabo.

Etapas	Algoritmo C3-1	Filtro de Información
Predicción	Absorción tipo (1)	$\hat{\mathbf{y}}_{t+1/t} = \mathbf{Y}_{t+1/t} (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \hat{\mathbf{y}}_{t/t} + \mathbf{u}_t)$ $\mathbf{Y}_{t+1/t} = (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \mathbf{F}_t^T + \mathbf{Q}_t)^{-1}$
Validación (opcional) de medida $\zeta_{s,t}$	$\Phi_{T_AUX}(\mathbf{x}_{t-1}, \mathbf{x}_t) = \frac{\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t)}{\prod_s \Phi_{\mathbf{x}_t\mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}, \mathbf{x}_t}(\mathbf{x}_t)}$ Realización de una absorción tipo 4, sin actualizar los potenciales del árbol, desde $\Phi_{T_AUX} \rightarrow \Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$ Marginalización de \mathbf{x}_t de $\Phi_{M_AUX} \rightarrow \Phi=(g,h,K)$ $\left. \begin{array}{l} \hat{\mathbf{r}} = \zeta_{s,t} - \mathbf{K}^{-1} \mathbf{h} \\ \mathbf{S} = \mathbf{K}^{-1} \end{array} \right\} \Rightarrow \text{Test: } \hat{\mathbf{r}}^T \mathbf{K} \hat{\mathbf{r}} \leq \gamma_s \quad ?$	$\hat{\mathbf{r}}_{s,t+1/t} = \zeta_{s,t+1} - \mathbf{H}_{s,t+1} \mathbf{Y}_{t+1/t}^{-1} \hat{\mathbf{y}}_{t+1/t}$ $\mathbf{S}_{s,t+1/t} = \mathbf{H}_{s,t+1} \mathbf{Y}_{t+1/t}^{-1} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ Test: $\hat{\mathbf{r}}_{s,t+1/t}^T \mathbf{S}_{s,t+1/t}^{-1} \hat{\mathbf{r}}_{s,t+1/t} \leq \gamma_s \quad ?$
Asimilación	Para cada sensor: Incorporación de la medida como evidencia en su potencial de medida Absorción tipo (3)	Para cada sensor: $\mathbf{i}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}$ $\mathbf{I}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1}$ $\hat{\mathbf{y}}_{t+1/t+1} = \hat{\mathbf{y}}_{t+1/t} + \sum_{s=1}^S \mathbf{i}_{s,t+1}$ $\mathbf{Y}_{t+1/t+1} = \mathbf{Y}_{t+1/t} + \sum_{s=1}^S \mathbf{I}_{s,t+1}$

Tabla C3-2. Operaciones necesarias para realizar una etapa del filtro de Información y del algoritmo de inferencia exacto C3-1 con una ventana temporal $n=1$. Se supone que las medidas llegan de forma temporalmente ordenada dentro de su periodo de muestreo. Para comparar la complejidad hay que tener en cuenta que cada operación de absorción está formada por tres operaciones: la marginalización de una variable del potencial origen, la asignación del valor obtenido en la marginalización al potencial de un separador, y la actualización del potencial destino por medio del producto y cociente de los valores antiguo y nuevo del potencial separador, respectivamente.

Algoritmo C3-1	Filtro de Información
$\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) \rightarrow$ escalar, vector $2n_x$, matriz $2n_x \times 2n_x$ $\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{z}_{s,t}}(\mathbf{x}_t) \rightarrow$ escalar, vector n_x , matriz $n_x \times n_x$ Para cada sensor: $\Phi_{\mathbf{x}_t\mathbf{z}_{s,t}}(\mathbf{x}_t, \mathbf{z}_{s,t}) \rightarrow$ escalar, vector n_x+n_s , matriz $(n_x+n_s) \times (n_x+n_s)$ $\Phi_{\mathbf{x}_t\mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_t) \rightarrow$ escalar, vector n_x , matriz $n_x \times n_x$	$\hat{\mathbf{y}}_{t+1/t} \rightarrow$ vector n_x $\mathbf{Y}_{t+1/t} \rightarrow$ matriz $n_x \times n_x$
TOTAL (um): $2 + 2S + (3 + 2S)n_x + (5 + 2S)n_x^2 + (2n_x + 1) \sum_{s=1}^S n_s + \sum_{s=1}^S n_s^2$	TOTAL (um): $n_x + n_x^2$

Tabla C3-3. Memoria necesaria para almacenar las variables del filtro de Información y del algoritmo de inferencia exacto C3-1 con una ventana temporal $n=1$. Se supone que el vector del estado del sistema tiene n_x elementos y que cada uno de los S vectores de medida tiene n_s elementos. El consumo TOTAL de memoria ha sido calculado suponiendo que el escalar, y cada uno de los elementos de los vectores y matrices ocupan una unidad de memoria (um).

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

Por lo tanto, la única ventaja existente entre el algoritmo de inferencia exacto C3-1 y el filtro de Información es la capacidad del primero para asimilar la información sensorial que llega al mismo de forma temporalmente retardada y desordenada. En el análisis del capítulo anterior se demostró que esta capacidad es debida a:

- que el algoritmo de inferencia exacta almacena en distintas componentes del vector \mathbf{h} y de la matriz \mathbf{K} de los potenciales de transición de estados $\mathbf{x}_{t-1}\mathbf{x}_t$ la información sensorial $\zeta_{1:S,t}$ asociada a sus S sensores en el periodo de muestreo t , y la información sensorial $\zeta_{1:S,1:t-1}$ previa a t , y a
- que en las absorciones tipo (1) realizadas a través de los conglomerados $(\mathbf{x}_{t-1}\mathbf{x}_t$ y $\mathbf{x}_t\mathbf{x}_{t+1})$ y separadores $(\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1})$ de transición de estados se añade la información más reciente que se tiene sobre el estado anterior \mathbf{x}_{t-1} y se quita la información que se tenía antes de la llegada de la medida $\zeta_{s,t}$ sobre el estado anterior \mathbf{x}_{t-1} .

A partir de estas observaciones y con el objetivo de superar los inconvenientes del algoritmo de inferencia exacto C3-1, en esta sección se propondrán dos versiones del filtro de Información para estimar el estado de sistemas de control lineales y no lineales, cuyos sensores obtienen información sobre el sistema de manera asíncrona y la proporcionan al sistema de fusión multisensorial de forma temporalmente desordenada. Estos filtros podrán ser utilizados como los algoritmos de inferencia de la red bayesiana de la figura C3-1, sustituyendo en esta labor al algoritmo de inferencia exacta C3-1.

En la primera versión del filtro de Información se tratará el caso puramente lineal, que es el caso ya resuelto por el algoritmo de inferencia C3-1. A continuación se desarrollará la versión para el caso de sistemas de control con ecuaciones de transición de estados o ecuaciones de medida no lineales, utilizando como algoritmos de partida el filtro de Información obtenido para el caso puramente lineal y el filtro de Información Extendido básico. Las expresiones habituales de los diferentes filtros de Información se encuentran recogidas en el apéndice I. Por último se destacan las características relevantes de los filtros propuestos y éstos son comparados con otros algoritmos propuestos en la literatura para la estimación del estado de un sistema de control con medidas temporalmente desordenadas.

C3.2.1. Filtro de Información para sistemas de control lineales con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

En la etapa de asimilación del filtro de Información básico, cuyas ecuaciones se presentan en la siguiente figura, la información sobre el estado $\hat{\mathbf{y}}_{t/t}$ (y su covarianza $\mathbf{Y}_{t/t}$) en el instante t se obtiene

sumándole la información sensorial $\mathbf{i}_{s,t}$ (y su covarianza $\mathbf{I}_{s,t}$) proporcionada por cada uno de los sensores s en dicho instante, a la información sobre el estado $\hat{\mathbf{y}}_{t/t-1}$ (y a su covarianza, $\mathbf{Y}_{t/t-1}$) obtenida a partir de un etapa de predicción sobre la información (y covarianza) que se tenía en el instante de muestreo anterior $t-1$. Por lo tanto, es posible descomponer las ecuaciones en dos términos aditivos, uno responsable de acumular la información sensorial del instante de muestreo actual y otro encargado de almacenar la información del estado, estimada a partir de toda la información sensorial obtenida en instantes de muestreo anteriores.

$$\begin{array}{ccc}
 \begin{array}{|c|} \hline \hat{\mathbf{y}}_{t/t} \\ \hline \mathbf{Y}_{t/t} \\ \hline \text{Información del} \\ \text{estado con la} \\ \text{información} \\ \text{sensorial hasta } t \\ \hline \end{array} & = & \begin{array}{|c|} \hline \hat{\mathbf{y}}_{t/t-1} \\ \hline \mathbf{Y}_{t/t-1} \\ \hline \text{Información del} \\ \text{estado con la} \\ \text{información} \\ \text{sensorial} \\ \text{anterior a } t \\ \hline \end{array} + \begin{array}{|c|} \hline \sum_{s=1}^S \mathbf{i}_{s,t} \\ \hline \sum_{s=1}^S \mathbf{I}_{s,t} \\ \hline \text{Información} \\ \text{sensorial para } t \\ \hline \end{array}
 \end{array}$$

Figura C3-3. Descomposición de la expresión de asimilación del filtro de Información en un término que contiene toda la información sensorial del instante de muestreo t y otra que contiene la información del estado con la información sensorial hasta un instante de muestreo anterior.

Esta descomposición, esquematizada en la figura C3-3, permite almacenar en variables diferentes los valores de cada uno de los dos términos aditivos de la expresión de asimilación, hecho que sustentará la asimilación temporalmente desordenada y retardada de la información sensorial, por medio de un proceso equivalente al que el algoritmo de inferencia exacto C3-1 lleva a cabo automáticamente.

La nomenclatura elegida para las variables asociadas a cada uno de los términos aditivos es la siguiente:

- \mathbf{i}_t (e \mathbf{I}_t) almacenará toda la información sensorial (y su covarianza) asociada al instante t que ya ha sido recibida por el algoritmo.
- $\hat{\mathbf{y}}_{t/AASIM \leq t-1}$ (e $\mathbf{Y}_{t/AASIM \leq t-1}$) almacenará la información (y su covarianza) que se tiene sobre el estado hasta el instante t , obtenida tras una predicción de la información obtenida sobre el instante $t-1$. $AASIM \leq t-1$, abreviatura de All the Available Sensor Information Measured Until $t-1$, se utiliza para hacer referencia a toda la información sensorial disponible por el algoritmo que ha sido medida hasta el instante $t-1$.

El resultado de la asimilación se almacenará temporalmente en las variables $\hat{\mathbf{y}}_{t/AASIM \leq t}$ e $\mathbf{Y}_{t/AASIM \leq t}$. La notación elegida es semejante a la del filtro de Información tradicional, salvo por la inclusión del termino $AASIM \leq t$ para representar que no es toda la información sensorial para instantes de

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

muestreo anteriores o iguales a t , sino la información sensorial para instantes de muestreo anteriores o iguales a t que ha sido recibida por el algoritmo. De esta forma, al igual que $\hat{y}_{t|t-1}$ almacenaba el valor de la información del estado en el instante t tras la etapa de predicción, e $\hat{y}_{t|t}$ recogía el valor de la información del estado en el instante t tras la etapa de asimilación, $\hat{y}_{t|AASIM \leq t-1}$ almacenará el valor de la información del estado en el instante t tras la etapa de predicción con toda la información sensorial medida antes de t que ya ha sido asimilada por el algoritmo, e $\hat{y}_{t|AASIM \leq t}$ recogerá el valor de la información del estado en el instante t con toda la información sensorial medida hasta t que ya ha sido asimilada por el algoritmo.

La asimilación temporalmente desordenada para el filtro de Información se realizará, tan pronto como la medida $\zeta_{s,k}$ realizada en el instante k está disponible (en un instante posterior t) y se haya comprobado que es válida, por medio de las siguientes operaciones:

- La medida realizada se convierte en la información sensorial $i_{s,k}$ (e $I_{s,k}$) y se añade al resto de la información sensorial ya asimilada para dicho periodo de muestreo, que se encuentra almacenada en las variables i_k (e I_k).
- Para obtener la información sobre el estado del sistema $\hat{y}_{k|AASIM \leq k}$ (e $Y_{k|AASIM \leq k}$), que tiene que incorporar toda la información sensorial capturada hasta el instante k que ya ha sido recibida y procesada por el filtro de Información, se añade la información sensorial i_k (e I_k) disponible y capturada en el instante k , a la información del estado $\hat{y}_{k|AASIM \leq k-1}$ (e $Y_{k|AASIM \leq k-1}$) que almacena toda la información sensorial disponible y capturada en instantes previos a k .
- La variable temporal $\hat{y}_{k|AASIM \leq k}$ (e $Y_{k|AASIM \leq k}$) es utilizada como punto de partida de una etapa de predicción del filtro de Información, de la que se obtiene el nuevo valor para la variable $\hat{y}_{k+1|AASIM \leq k}$ (e $Y_{k+1|AASIM \leq k}$), que constituye el primer término aditivo de la ecuación de asimilación del periodo de muestreo $k+1$.
- Acto seguido se suma el nuevo valor de $\hat{y}_{k+1|AASIM \leq k}$ (e $Y_{k+1|AASIM \leq k}$) a la información sensorial i_{k+1} e I_{k+1} , que ha sido capturada en el instante $k+1$ y que ya ha sido asimilada anteriormente por el algoritmo, y se realiza una nueva predicción, obteniéndose los nuevos valores de $\hat{y}_{k+2|AASIM \leq k+1}$ e $Y_{k+2|AASIM \leq k+1}$.
- Esta última etapa, formada por una suma y una predicción, se realizará de forma sucesiva hasta que se actualice el valor de la información para el instante de muestreo actual del filtro de Información t .

Como la etapa de predicción necesita el valor de la señal de control u_k es necesario almacenar también los últimos n valores de la señal de control, para poder realizar las predicciones sucesivas asociadas a la asimilación de medidas temporalmente retardadas.

Si se quiere implementar una etapa de validación basada en la estimación del valor de la medida s para el instante de muestreo k , se utilizarán las expresiones del filtro de Información encargadas de realizar esta etapa opcional, tomando como valor estimado de la información del estado (y su covarianza correspondiente) el almacenado en la variable $\hat{y}_{k|AASIM \leq k-1}$ (e $Y_{k|AASIM \leq k-1}$). De esta forma se consigue, para los casos en los que las medidas lleguen al filtro de Información de forma temporalmente desordenada, que dentro de un mismo periodo de muestreo k la información sensorial de una medida no influya en el proceso de validación de medidas de otros sensores que se encuentren disponibles más tarde. Es decir, que ante el comportamiento esquematizado en la sección c) de la figura C3-4, las medidas de diferentes sensores realizadas en un instante t no sean utilizadas para validar las medidas de otros tomadas en el mismo instante, pero que lleguen al algoritmo posteriormente.

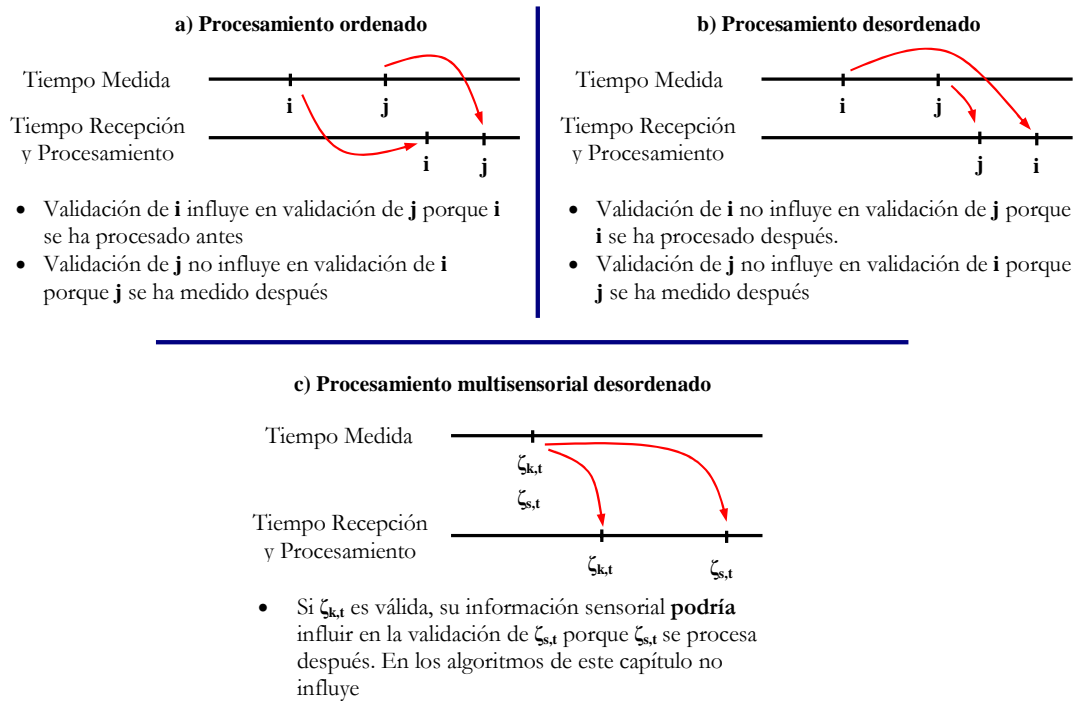


Figura C3-4. Influencia en el proceso de validación del orden de procesamiento de las medidas temporalmente ordenadas y desordenadas.

Sin embargo, el problema que existe ante un posible desorden temporal en la validación entre diferentes sensores y distintos instantes de medida no ha sido resuelto por el algoritmo. El problema se encuentra esquematizado en las regiones a) y b) de la figura C3-4. Para resolverlo, a la hora de realizar una validación se utilizará únicamente la información sensorial de instantes de muestreo anteriores que ya haya sido recibida por el filtro de Información. De esta forma, si dos medidas realizadas por dos sensores en los instantes i y j , con j mayor que i , llegan temporalmente desordenadas al algoritmo de inferencia, es decir la medida j antes de la i , la validación de la

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

medida **j** no utilizará la información sensorial de la medida **i** al no haber sido ésta asimilada todavía en el valor estimado del estado del sistema. Si se volviese a realizar el test de validación de la medida **j** una vez que el algoritmo hubiese asimilado la medida **i**, podría darse el caso que el nuevo resultado del test fuese diferente.

Al igual que en el capítulo anterior, se mantendrá el resultado del proceso de validación una vez que este haya sido realizado, porque si el filtro está bien sintonizado se supone que los cambios que se podrían producir revalidando las medidas no influirán significativamente en el resultado final del algoritmo, y dicha revalidación sobrecargaría el sistema, además de no permitirnos ir almacenando toda la información sensorial proporcionada por los diferentes sensores para cada instante de muestreo **k** en las variables **i_k** e **I_k**.

A continuación se recoge la nueva versión propuesta para el filtro de Información para sistemas de control lineales multisensoriales, cuyos sensores realizan medidas de forma asíncrona y éstas pueden estar disponibles de forma temporalmente desordenada. En la misma se utiliza el test chi-cuadrado como test de validación.

- 1) Inicialización
 - a. Obtener el modelo del sistema (ecuación de transición de estados) y de los sensores (ecuaciones de medida).
 - b. Fijar el tamaño de la ventana de tiempo **n**.
 - c. **t=1**
 - d. Inicializar las variables asociadas al instante **t=1**

$$\mathbf{Y}_{I/AASIM \leq 0} = (\mathbf{F}_0 \mathbf{P}_0 \mathbf{F}_0^T + \mathbf{Q}_0)^{-1}$$

$$\hat{\mathbf{y}}_{I/AASIM \leq 0} = \mathbf{Y}_{I/AASIM \leq 0} (\mathbf{F}_0 \hat{\mathbf{x}}_0 + \mathbf{u}_0)$$

$$\mathbf{i}_I = \mathbf{0}, \mathbf{I}_I = \mathbf{0}$$
- 2) Bucle principal: **while(true)**
 - a. If ha pasado el tiempo de un muestreo, hay que inicializar las variables del nuevo instante, y posiblemente eliminar las variables del periodo más antiguo:
 - i. **t=t+1**
 - ii. Se asimila la información del instante anterior **t-1**

$$\hat{\mathbf{y}}_{t-1|AASIM \leq t-1} = \hat{\mathbf{y}}_{t-1|AASIM \leq t-2} + \mathbf{i}_{t-1}$$

$$\mathbf{Y}_{t-1|AASIM \leq t-1} = \mathbf{Y}_{t-1|AASIM \leq t-2} + \mathbf{I}_{t-1}$$
 - iii. Se inicializan las variables para el instante **t**

$$\mathbf{Y}_{t|AASIM \leq t-1} = (\mathbf{F}_{t-1} \mathbf{Y}_{t-1|AASIM \leq t-1}^T + \mathbf{Q}_{t-1})^{-1}$$

$$\hat{\mathbf{y}}_{t|AASIM \leq t-1} = \mathbf{Y}_{t|AASIM \leq t-1} (\mathbf{F}_{t-1} \mathbf{Y}_{t-1|AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t-1|AASIM \leq t-1} + \mathbf{u}_{t-1})$$

$$\mathbf{i}_t = \mathbf{0}, \mathbf{I}_t = \mathbf{0}$$

<p>iv. Si $t-n > 0$ hay que eliminar los elementos asociados al instante $t-n$, es decir las variables:</p> $\mathbf{Y}_{t-n AASIM \leq t-n-1}, \hat{\mathbf{y}}_{t-n AASIM \leq t-n-1}, \mathbf{i}_{t-n}, \mathbf{I}_{t-n}, \mathbf{u}_{t-n}$ <p>b. Si hay una medida $\zeta_{s,k}$ disponible, hay que comprobar si ésta se ha realizado en un k que este dentro de la ventana temporal n.</p> <p>i. Si no ($k > t-n$) la medida no se puede asimilar, por lo que es conveniente mandar un mensaje al usuario para que detecte que con el n elegido se pierden medidas</p> <p>ii. Si ($k > t-n$) la medida se podrá asimilar, siempre que sea válida:</p> <p>1. Obtener el valor estimado de la medida</p> $\hat{\mathbf{z}}_{s,k AASIM \leq k-1} = \mathbf{H}_{s,k} \mathbf{Y}_{k AASIM \leq k-1}^{-1} \hat{\mathbf{y}}_{k AASIM \leq k-1}$ $\mathbf{S}_{s,k AASIM \leq k-1} = \mathbf{H}_{s,k} \mathbf{Y}_{k AASIM \leq k-1}^{-1} \mathbf{H}_{s,k}^T + \mathbf{R}_{s,k}$ <p>2. Realizar el test chi-cuadrado de validación basado en el valor estimado de la medida</p> $\hat{\mathbf{r}}_{s,k AASIM \leq k-1} = \zeta_{s,k} - \hat{\mathbf{z}}_{s,k AASIM \leq k-1}$ <p>Test: ¿ $\hat{\mathbf{r}}_{s,k AASIM \leq k-1}^T \mathbf{S}_{s,k AASIM \leq k-1}^{-1} \hat{\mathbf{r}}_{s,k AASIM \leq k-1} \leq \gamma_s$?</p> <p>3. Si la medida es válida:</p> <p>a. Obtener la información sensorial asociada a la medida $\zeta_{s,k}$:</p> $\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \zeta_{s,k}$ $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}$ <p>b. Añadir la nueva información sensorial al resto de la información sensorial asociada a k:</p> $\mathbf{i}_k = \mathbf{i}_k + \mathbf{i}_{s,k}$ $\mathbf{I}_k = \mathbf{I}_k + \mathbf{I}_{s,k}$ <p>c. Si $k \neq t$, hay que propagar la nueva información sensorial hacia delante, de forma que se actualicen el término aditivo de la etapa de asimilación que incluye la información sobre el estado en instante de muestreo anteriores:</p> <p>for $j = k : t-1$</p> $\hat{\mathbf{y}}_{j AASIM \leq j} = \hat{\mathbf{y}}_{j AASIM \leq j-1} + \mathbf{i}_j$ $\mathbf{Y}_{j AASIM \leq j} = \mathbf{Y}_{j AASIM \leq j-1} + \mathbf{I}_j$ $\mathbf{Y}_{j+1 AASIM \leq j} = (\mathbf{F}_j \mathbf{Y}_{j AASIM \leq j}^{-1} \mathbf{F}_j^T + \mathbf{Q}_j)^{-1}$ $\hat{\mathbf{y}}_{j+1 AASIM \leq j} = \mathbf{Y}_{j+1 AASIM \leq j} (\mathbf{F}_j \mathbf{Y}_{j AASIM \leq j}^{-1} \hat{\mathbf{y}}_{j AASIM \leq j} + \mathbf{u}_j)$ <p>end</p>
--

Algoritmo C3-2. Filtro de Información para un sistema lineal de control con múltiples sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente retardada y desordenada.

Esta nueva versión del filtro de Información es computacionalmente equivalente al algoritmo de inferencia exacto C3-1, aunque su consumo de memoria y complejidad computacional es menor. En las tres tablas siguientes se presenta una comparativa de los dos algoritmos, desde el punto de

C3.2. *Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada*

vista de sus necesidades de memoria y de cómputo. Asimismo, en el capítulo siguiente se comparan los resultados obtenidos por los dos algoritmos sobre un ejemplo puramente lineal.

Etapa	Algoritmo C3-1	Filtro de Información C3-2
Nuevo instante de muestreo $t=t+1$	$\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) =$ $= \left(g, \begin{bmatrix} -\mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \mathbf{u}_{t-1} \\ \mathbf{Q}_{t-1}^{-1} \mathbf{u}_{t-1} \end{bmatrix}, \begin{bmatrix} \mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \mathbf{F}_{t-1} & -\mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \\ -\mathbf{Q}_{t-1}^{-1} \mathbf{F}_{t-1} & \mathbf{Q}_{t-1}^{-1} \end{bmatrix} \right)$ $\Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}(\mathbf{x}_t, \mathbf{z}_{s,t}) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & -\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right)$ $\Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1} \mathbf{x}_t}(\mathbf{x}_t) = (\emptyset, [\mathbf{0}], [\mathbf{0}])$ $\Phi_{\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}}(\mathbf{x}_t) = (\emptyset, [\mathbf{0}], [\mathbf{0}])$ $\Phi_{\mathbf{x}_{t-1} \mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = \Phi_{\mathbf{x}_{t-1} \mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) \cdot \Phi_{\mathbf{x}_{t-2} \mathbf{x}_{t-1} \cap \mathbf{x}_{t-1} \mathbf{x}_t}(\mathbf{x}_{t-1})$ $\Phi_{\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}}(\mathbf{x}_t) = \int \Phi_{\mathbf{x}_{t-1} \mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) d\mathbf{x}_{t-1}$	$\hat{\mathbf{y}}_{t-1 AASIM \leq t-1} = \hat{\mathbf{y}}_{t-1 AASIM \leq t-2} + \mathbf{i}_{t-1}$ $\mathbf{Y}_{t-1 AASIM \leq t-1} = \mathbf{Y}_{t-1 AASIM \leq t-2} + \mathbf{I}_{t-1}$ $\mathbf{Y}_{t AASIM \leq t-1} = (\mathbf{F}_{t-1} \mathbf{Y}_{t-1 AASIM \leq t-1}^T \mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})^{-1}$ $\hat{\mathbf{x}}_{t-1 AASIM \leq t-1} = \mathbf{Y}_{t-1 AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t-1 AASIM \leq t-1}$ $\hat{\mathbf{y}}_{t AASIM \leq t-1} = \mathbf{Y}_{t AASIM \leq t-1} \cdot$ $\cdot (\mathbf{F}_{t-1} \hat{\mathbf{x}}_{t-1 AASIM \leq t-1} + \mathbf{u}_{t-1})$ $\mathbf{i}_t = \mathbf{0}$ $\mathbf{I}_t = \mathbf{0}$
	Si $t > n$ eliminar los potenciales asociados al instante $t-n$ (1 de transición, 1 del separador de transición, s de medida, s de los separadores de medida)	Si $t > n$ eliminar las variables de ese instante: $\mathbf{i}_{t-n}, \mathbf{I}_{t-n}, \mathbf{Y}_{t-n AASIM \leq t-n-1}, \hat{\mathbf{y}}_{t-n AASIM \leq t-n-1}, \mathbf{u}_{t-n}$
Medida disponible $\zeta_{s,k}$	Validación (opcional)	
	Asimilación	
	$\Phi_{T_AUX}(\mathbf{x}_{k-1}, \mathbf{x}_k) = \frac{\Phi_{\mathbf{x}_{k-1}\mathbf{x}_k}(\mathbf{x}_{k-1}, \mathbf{x}_k)}{\prod_s \Phi_{\mathbf{x}_k \mathbf{z}_{s,k} \cap \mathbf{x}_{k-1} \mathbf{x}_k}(\mathbf{x}_k)}$ <p>Realizar una absorción tipo 4, sin actualizar los potenciales del árbol: $\Phi_{T_AUX} \rightarrow \Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$</p> <p>Marginalizar \mathbf{x}_t de $\Phi_{M_AUX} \rightarrow \Phi=(g,h,K)$</p> $\hat{\mathbf{r}} = \zeta_{s,t} - \mathbf{K}^{-1} \mathbf{h}$ $\mathbf{S} = \mathbf{K}^{-1}$ $\Rightarrow \text{Test: } \hat{\mathbf{r}}^T \mathbf{K} \hat{\mathbf{r}} \leq \gamma_s ?$	$\hat{\mathbf{z}} = \mathbf{H}_{s,k} \mathbf{Y}_{k AASIM \leq k-1}^{-1} \hat{\mathbf{y}}_{k AASIM \leq k-1}$ $\mathbf{S} = \mathbf{H}_{s,k} \mathbf{Y}_{k AASIM \leq k-1}^{-1} \mathbf{H}_{s,k}^T + \mathbf{R}_{s,k}$ $\hat{\mathbf{r}} = \zeta_{s,k} - \hat{\mathbf{z}}$ <p>Test: $\hat{\mathbf{r}}^T \mathbf{S}^{-1} \hat{\mathbf{r}} \leq \gamma_s ?$</p>
	<p>Introducir la medida $\zeta_{k,t}$ como evidencia en el nodo sensorial $\mathbf{x}_{kZs,kk}$</p> <p>Realizar una absorción tipo (3): desde el nodo de transición de estados asociado al nodo sensorial sobre el que se ha medido (absorción $\mathbf{x}_{kZs,k} \rightarrow \mathbf{x}_{k-1}\mathbf{x}_k$)</p> <p>Si $k \neq t$, realizar absorciones tipo (1) de forma sucesiva y ordenada de izquierda a derecha desde el nodo de transición de estados asociado a la medida disponible hasta el último (primera absorción $\mathbf{x}_{k-1}\mathbf{x}_k \rightarrow \mathbf{x}_k \mathbf{x}_{t+1}, \dots$, y última absorción $\mathbf{x}_{t-2}\mathbf{x}_{t-1} \rightarrow \mathbf{x}_{t-1}\mathbf{x}_t$)</p> <p>Actualizar el valor del potencial del separador $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}$ a partir de una marginalización sobre el potencial del conglomerado $\mathbf{x}_{t-1}\mathbf{x}_t$:</p> $\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}}(\mathbf{x}_t) = \int \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) d\mathbf{x}_{t-1}$	$\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \zeta_{s,k}$ $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}$ $\mathbf{i}_k = \mathbf{i}_k + \mathbf{i}_{s,k}$ $\mathbf{I}_k = \mathbf{I}_k + \mathbf{I}_{s,k}$ <p>for $j = k : t-1$</p> $\hat{\mathbf{y}}_{j AASIM \leq j} = \hat{\mathbf{y}}_{j AASIM \leq j-1} + \mathbf{i}_j$ $\mathbf{Y}_{j AASIM \leq j} = \mathbf{Y}_{j AASIM \leq j-1} + \mathbf{I}_j$ $\mathbf{Y}_{j+1 AASIM \leq j} = (\mathbf{F}_j \mathbf{Y}_{j AASIM \leq j}^T \mathbf{F}_j^T + \mathbf{Q}_j)^{-1}$ $\hat{\mathbf{y}}_{j+1 AASIM \leq j} = \mathbf{Y}_{j+1 AASIM \leq j} \cdot$ $\cdot (\mathbf{F}_j \mathbf{Y}_{j AASIM \leq j}^{-1} \hat{\mathbf{y}}_{j AASIM \leq j} + \mathbf{u}_j)$ <p>end</p>

Tabla C3-4. Operaciones necesarias para el bucle principal del algoritmo para el filtro de Información C3-2 y del algoritmo de inferencia C3-1.

	Algoritmo C3-1	Filtro de Información C3-2
Etapa de Inicialización (t=1)	<p>Obtener el modelo del sistema (ecuación de transición de estados) y de los sensores (ecuaciones de medida).</p> <p>Fijar el tamaño de la ventana de tiempo n</p>	
	$\Phi_{\mathbf{x}_0 \mathbf{x}_I}(\mathbf{x}_0, \mathbf{x}_I) = \left(g, \begin{bmatrix} \hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \\ \mathbf{Q}_0^{-1} \mathbf{u}_0 \end{bmatrix}, \begin{bmatrix} \hat{\mathbf{Y}}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 & -\mathbf{F}_0^T \mathbf{Q}_0^{-1} \\ -\mathbf{Q}_0^{-1} \mathbf{F}_0 & \mathbf{Q}_0^{-1} \end{bmatrix} \right)$ $\Phi_{\mathbf{x}_I \mathbf{z}_{s,I}}(\mathbf{x}_I, \mathbf{z}_{s,I}) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,I}^T \mathbf{R}_{s,I}^{-1} \mathbf{H}_{s,I} & -\mathbf{H}_{s,I}^T \mathbf{R}_{s,I}^{-1} \\ -\mathbf{R}_{s,I}^{-1} \mathbf{H}_{s,I} & \mathbf{R}_{s,I}^{-1} \end{bmatrix} \right)$ $\Phi_{\mathbf{x}_I \mathbf{z}_{s,I} \mathbf{v}_{I,I} \cap \mathbf{x}_0 \mathbf{x}_I}(\mathbf{x}_I) = (\emptyset, [\mathbf{0}], [\mathbf{0}]) \quad \Phi_{\mathbf{x}_0 \mathbf{x}_I \cap \mathbf{x}_I \mathbf{x}_I}(\mathbf{x}_I) = (\emptyset, [\mathbf{0}], [\mathbf{0}])$ $\Phi_{\mathbf{x}_0 \mathbf{x}_I \cap \mathbf{x}_I \mathbf{x}_I}(\mathbf{x}_I) = \int \Phi_{\mathbf{x}_0 \mathbf{x}_I}(\mathbf{x}_0, \mathbf{x}_I) d\mathbf{x}_0$	$\mathbf{Y}_{I/AASIM \leq 0} = (\mathbf{F}_0 \mathbf{P}_0 \mathbf{F}_0^T + \mathbf{Q}_0)^{-1}$ $\hat{\mathbf{y}}_{I/AASIM \leq 0} = \mathbf{Y}_{I/AASIM \leq 0} (\mathbf{F}_0 \hat{\mathbf{x}}_0 + \mathbf{u}_0)$ $\mathbf{i}_I = \mathbf{0}$ $\mathbf{I}_I = \mathbf{0}$

Tabla C3-5. Operaciones necesarias para la etapa de inicialización del filtro de Información C3-2 y del algoritmo de inferencia C3-1.

Algoritmo de Inferencia Exacto C3-1	Filtro de Información C3-2
<p>Para cada instante de tiempo se tiene:</p> <p>$\Phi_{\mathbf{x}_{t-1} \mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) \rightarrow$ escalar, vector $2n_x$, matriz $2n_x \times 2n_x$</p> <p>$\Phi_{\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}}(\mathbf{x}_t) \rightarrow$ escalar, vector n_x, matriz $n_x \times n_x$</p> <p>Para cada instante de tiempo y sensor se tiene:</p> <p>$\Phi_{\mathbf{x}_t \mathbf{z}_{s,t}}(\mathbf{x}_t, \mathbf{z}_{s,t}) \rightarrow$ escalar, vector $n_x + n_s$, matriz $(n_x + n_s) \times (n_x + n_s)$</p> <p>$\Phi_{\mathbf{x}_t \mathbf{z}_{s,t} \cap \mathbf{x}_{t-1} \mathbf{x}_t}(\mathbf{x}_t) \rightarrow$ escalar, vector n_x, matriz $n_x \times n_x$</p>	<p>Para cada instante de tiempo se tiene:</p> <p>$\hat{\mathbf{y}}_{t/AASIM \leq t-1} \rightarrow$ vector n_x</p> <p>$\mathbf{Y}_{t/AASIM \leq t-1} \rightarrow$ matriz $n_x \times n_x$</p> <p>$\mathbf{i}_t \rightarrow$ vector n_x</p> <p>$\mathbf{I}_t \rightarrow$ matriz $n_x \times n_x$</p> <p>$\mathbf{u}_t \rightarrow$ vector n_x</p>
<p>TOTAL (um) :</p> $V \times \left(2 + 2S + (3 + 2S)n_x + (5 + 2S)n_x^2 + (2n_x + 1) \sum_{s=1}^S n_s + \sum_{s=1}^S n_s^2 \right)$	<p>TOTAL (um):</p> $V \times (3n_x + 2n_x^2)$

Tabla C3-6. Memoria necesaria para almacenar para cada instante de muestreo dentro de la ventana temporal las variables del filtro de Información C3-2 y del algoritmo de inferencia exacto C3-1. Se supone que el estado del sistema es un vector de dimensión n_x , y que para cada uno de los S sensores devuelve un vector de medida de dimensión n_s . El consumo TOTAL de memoria ha sido calculado suponiendo que el escalar, y cada uno de los elementos de los vectores y matrices ocupan una unidad de memoria (um), y que el tamaño de la ventana de tiempo es V.

C3.2.2. Filtro de Información Mixto para sistemas de control no lineales con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

La versión del filtro de Información recogida en el algoritmo C3-2 es únicamente válida para sistema de control modelables con funciones lineales, tanto para el caso de la función de transición de estados como para las funciones que relacionan las medidas con el estado del sistema. Esta restricción, establecida en el capítulo anterior como consecuencia del algoritmo de inferencia exacto elegido, puede ser evitada si éste es sustituido por otro que permita realizar inferencias sobre sistemas gaussianos no lineales.

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

El filtro de Información Extendido básico, cuyas ecuaciones para el caso de sistemas multisensoriales se recogen en la tabla A1.9 del apéndice I, es la versión del el algoritmo de Información para el caso no lineal. Su funcionamiento es semejante al del filtro de Información básico, aunque las ecuaciones utilizadas para la predicción, validación y asimilación son ligeramente diferentes. Por lo tanto, la versión del filtro de Información Extendido para sistemas de control no lineales cuyas medidas pueden estar temporalmente desordenadas se obtendrá introduciendo en el filtro de Información C3-2 las modificaciones que la no linealidad produce en las etapas de predicción, validación y asimilación del filtro de Información Extendido. En las dos secciones siguientes se analizan por separado los cambios que hay que realizar en el algoritmo C3-2 debido a la no linealidad en la función de transición de estado y los debidos a la no linealidad de las funciones de medida. Después de dicho análisis se propondrá una nueva versión del filtro de Información para estimar el estado de un sistema con múltiples sensores, cuya función de transición pueda ser lineal o no lineal, y donde cada una de las funciones de medida pueda ser también lineal o no lineal. Es decir, el filtro funcionará como un filtro de Información para sistemas puramente lineales, como un filtro de Información Extendido para el caso puramente no lineal, y un filtro de Información Mixto cuando se combinan funciones lineales y no lineales.

▪ **No linealidad en la función de transición: $f(x,u,t)$**

La no linealidad en la función de transición de estados introduce dos cambios significativos, que pueden ser observados en la tabla comparativa C3-7, en la etapa de predicción del filtro:

- para obtener el nuevo valor de la información sobre el estado $\hat{\mathbf{y}}_{t+1|AASIM \leq t}$, utiliza la función no lineal $f(x,u,t)$, y
- a la hora de obtener el valor de la covarianza de la información sobre el estado $\mathbf{Y}_{t+1|AASIM \leq t}$ hay que obtener la matriz \mathbf{F}_t como el jacobiano respecto a la variable de estado de la función no lineal de transición.

Filtro de Información	Filtro de Información Extendido
$\mathbf{Y}_{t+1 AASIM \leq t} = \left(\mathbf{F}_t \mathbf{Y}_{t AASIM \leq t}^{-1} \mathbf{F}_t^T + \mathbf{Q}_t \right)^{-1}$ $\hat{\mathbf{y}}_{t+1 AASIM \leq t} = \mathbf{Y}_{t+1 AASIM \leq t} \left(\mathbf{F}_t \mathbf{Y}_{t AASIM \leq t}^{-1} \hat{\mathbf{y}}_{t AASIM \leq t} + \mathbf{u}_t \right)$	$\hat{\mathbf{x}}_{t AASIM \leq t} = \mathbf{Y}_{t AASIM \leq t}^{-1} \hat{\mathbf{y}}_{t AASIM \leq t}$ $\mathbf{F}_t = (\mathbf{J}f)_{\mathbf{x}}(\hat{\mathbf{x}}_{t AASIM \leq t}, \mathbf{u}_t, t)$ $\mathbf{Y}_{t+1 AASIM \leq t} = \left(\mathbf{F}_t \mathbf{Y}_{t AASIM \leq t}^{-1} \mathbf{F}_t^T + \mathbf{Q}_t \right)^{-1}$ $\hat{\mathbf{y}}_{t+1 AASIM \leq t} = \mathbf{Y}_{t+1 AASIM \leq t} f(\hat{\mathbf{x}}_{t AASIM \leq t}, \mathbf{u}_t, t)$

Tabla C3-7. Expresiones para la etapa de predicción del filtro de Información y del filtro de Información Extendido.

Tanto la función de transición $f(x,u,t)$ como su jacobiano $(\mathbf{J}f)_{\mathbf{x}}(x,u,t)$ son evaluados sobre el estado del sistema y no sobre la información del estado, por lo que para reducir el número de cálculos es

conveniente obtener primero dicho valor, almacenarlo en una variable temporal $\hat{\mathbf{x}}_{t|AASIM \leq t}$, y utilizarlo a la hora de evaluar las dos funciones. Esta conversión de espacio de trabajo no sobrecarga computacionalmente la etapa de predicción no lineal, ya que también era realizada en la etapa lineal dentro del cálculo del nuevo valor de la $\hat{\mathbf{y}}_{t+1|AASIM \leq t}$. Además, es importante recordar que la operación $(\mathbf{F}_t \mathbf{Y}_{t|AASIM \leq t}^{-1} \hat{\mathbf{y}}_{t|AASIM \leq t} + \mathbf{u}_t)$ es la evaluación de la función lineal de transición sobre el valor estimado del estado del sistema.

Por lo tanto, los cálculos adicionales en la predicción de la versión extendida del filtro de Información son los producidos a la hora de obtener el jacobiano y los debidos a la diferencia computacional existente entre la evaluación de la función de transición no lineal y la evaluación de la función de transición lineal $(\mathbf{F}_t \mathbf{x}_t + \mathbf{u}_t)$.

Por último, destacar que la etapa de predicción es utilizada en los pasos 2.a.iii y 2.b.iii.c del filtro de Información C3-2, y que en ambos casos antes de llevar a cabo la predicción se realiza una asimilación, por lo que las variables de partida $(\hat{\mathbf{y}}_{j|AASIM \leq j}$ e $\mathbf{Y}_{j|AASIM \leq j}$) de la predicción extendida ya están disponibles.

▪ **No linealidad en las funciones de medida $h_s(\mathbf{x}, t)$**

La no linealidad en las funciones de medida introduce varios cambios, que pueden ser observados en la tabla comparativa C3-8, en las expresiones utilizadas para la obtención de la información sensorial $\mathbf{i}_{s,t+1}$ (y su covarianza $\mathbf{I}_{s,t+1}$) que se realizan antes de la etapa de asimilación.

Filtro de Información	Filtro de Información Extendido
$\mathbf{i}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}$ $\mathbf{I}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1}$	$\hat{\mathbf{x}}_{t+1 AASIM \leq t} = \mathbf{Y}_{t+1 AASIM \leq t}^{-1} \hat{\mathbf{y}}_{t+1 AASIM \leq t}$ $\mathbf{H}_{s,t+1} = (\mathbf{J}h_s)_{\mathbf{x}}(\hat{\mathbf{x}}_{t+1 AASIM \leq t}, t+1)$ $\zeta_{s,t+1}^C = \zeta_{s,t+1} - h_s(\hat{\mathbf{x}}_{t+1 AASIM \leq t}, t+1) + \mathbf{H}_{s,t+1} \hat{\mathbf{x}}_{t+1 AASIM \leq t}$ $\mathbf{i}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}^C$ $\mathbf{I}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1}$

Tabla C3-8. Expresiones para el cálculo de la información sensorial para la etapa de asimilación del filtro de Información y del filtro de Información Extendido.

El cambio mas significativo en la etapa de asimilación entre el filtro de Información y su versión extendida, es que en el segundo caso aparece la necesidad de obtener el valor del estado $\hat{\mathbf{x}}_{t+1|AASIM \leq t}$ a partir del valor estimado sobre la información del estado $\hat{\mathbf{y}}_{t+1|AASIM \leq t}$ (y su covarianza $\mathbf{Y}_{t+1|AASIM \leq t}$) en la última etapa de predicción. Este valor es utilizado para obtener la matriz $\mathbf{H}_{s,t+1}$ (como la derivada primera de la función no lineal de medida) y para calcular el término corrector de la medida (como la diferencia existente entre el valor estimado por la medida a partir de la función no lineal $h_s(\mathbf{x}, t)$ y el valor estimado por la medida por la función linealizada $\mathbf{H}_{s,t+1} \hat{\mathbf{x}}$). Por lo

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

tanto, en el filtro de Información Extendido el valor de la información de la medida $\mathbf{i}_{s,t+1}$ (y su covarianza $\mathbf{I}_{s,t+1}$) dependen del valor estimado del estado en la etapa de predicción, a través del valor de la $\mathbf{H}_{s,t+1}$ y de la medida corregida $\zeta_{s,t+1}^C$.

En el filtro de Información C3-2, aunque el cálculo de la información de la medida $\mathbf{i}_{s,j}$ (e $\mathbf{I}_{s,j}$) únicamente se realiza en la etapa 2.b.iii.a, la asimilación propiamente dicha se realiza en las etapas 2.b.iii.b y 2.b.iii.c. En la primera, la información sensorial $\mathbf{i}_{s,k}$ (e $\mathbf{I}_{s,k}$) asociada a la medida $\zeta_{s,k}$ realizada en el instante \mathbf{k} que acaba de ser recibida por el algoritmo es añadida al resto de la información sensorial de dicho instante \mathbf{k} . En la segunda, se asimila la nueva información sensorial $\mathbf{i}_{s,k}$ (e $\mathbf{I}_{s,k}$) con la información sensorial de instantes posteriores \mathbf{j} (con \mathbf{j} mayor o igual a \mathbf{k}), a través de la realización sucesiva un conjunto de predicciones y asimilaciones.

Para el filtro de Información Extendido el cambio producido por la asimilación de una nueva medida $\zeta_{s,k}$ en el valor estimado para un instante posterior \mathbf{j} del estado $\hat{\mathbf{x}}_{j|AASIM \leq j-1}$ (y $\mathbf{P}_{j|AASIM \leq j-1}$), modificará el valor de la información sensorial $\mathbf{i}_{l,j}$ (e $\mathbf{I}_{l,j}$) asociada a la medida que cualquier sensor no lineal \mathbf{l} , debido a la dependencia existente entre la información sensorial y el valor del estado. Por lo tanto, para los sensores con funciones de medida no lineales es necesario recalcular la información sensorial $\mathbf{i}_{l,j}$ (e $\mathbf{I}_{l,j}$) antes de poder asimilarla, cada vez que se actualice mediante predicción el valor estimado de la información del estado $\hat{\mathbf{y}}_{j|AASIM \leq j-1}$ (e $\mathbf{Y}_{j|AASIM \leq j-1}$). Para poder recalcular la información sensorial $\mathbf{i}_{l,j}$ (y su covarianza $\mathbf{I}_{l,j}$) será necesario almacenar el valor original de las últimas medidas $\zeta_{l,m}$ del sensor \mathbf{l} que han sido realizadas dentro de la ventana temporal \mathbf{n} .

Es importante destacar que esta necesidad de recálculo de la información sensorial $\mathbf{i}_{l,j}$ (e $\mathbf{I}_{l,j}$) sobrecarga computacionalmente de forma significativa la nueva versión del filtro de Información Extendido. Aun más, cuando todos los sensores son no lineales impone el mismo gasto computacional al realizado al retrasar completamente el filtro de Información desde el instante de medida \mathbf{k} , asociado a la medida temporalmente desordenada $\zeta_{s,k}$, hasta el instante de muestreo \mathbf{t} , utilizando las medidas $\zeta_{s,j}$ realizadas en instantes \mathbf{j} de muestreo posteriores a \mathbf{k} pero que han sido recibidas antes por el filtro de Información Extendido.

La no linealidad en la función de medida también influye en la etapa de validación, hecho que se puede observar en la siguiente tabla comparativa.

Para el filtro de Información Extendido es necesario obtener la matriz $\mathbf{H}_{s,t+1}$ (como el jacobiano de la función no lineal $h_s(x,t)$) y en vez de estimar el valor de la medida con la expresión lineal $\mathbf{H}_{s,t+1}\hat{\mathbf{x}}_{t+1|AASIM \leq t}$ se utilizará la función de medida no lineal. En ambos filtros es necesario obtener el valor del estado del sistema $\hat{\mathbf{x}}_{t+1|AASIM \leq t}$ (y $\mathbf{P}_{t+1|AASIM \leq t}$) antes de estimar el valor de la medida.

Filtro de Información	Filtro de Información Extendido
$\hat{\mathbf{x}}_{t+1/AASIM \leq t} = \mathbf{Y}_{t+1/AASIM \leq t}^{-1} \hat{\mathbf{y}}_{t+1/AASIM \leq t}$ $\hat{\mathbf{r}} = \zeta_{s,t+1} - \mathbf{H}_{s,t+1} \hat{\mathbf{x}}_{t+1/AASIM \leq t}$ $\mathbf{S} = \mathbf{H}_{s,t+1} \mathbf{Y}_{t+1/t}^{-1} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ $\text{Test: } \hat{\mathbf{r}}^T \mathbf{S}^{-1} \hat{\mathbf{r}} \leq \gamma_s \quad ?$	$\hat{\mathbf{x}}_{t+1/AASIM \leq t} = \mathbf{Y}_{t+1/AASIM \leq t}^{-1} \hat{\mathbf{y}}_{t+1/AASIM \leq t}$ $\mathbf{H}_{s,t+1} = (\mathbf{J}h_s)_x(\hat{\mathbf{x}}_{t+1/AASIM \leq t}, t+1)$ $\hat{\mathbf{r}} = \zeta_{s,t+1} - h_s(\hat{\mathbf{x}}_{t+1/AASIM \leq t}, t+1)$ $\mathbf{S} = \mathbf{H}_{s,t+1} \mathbf{Y}_{t+1/t}^{-1} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ $\text{Test: } \hat{\mathbf{r}}^T \mathbf{S}^{-1} \hat{\mathbf{r}} \leq \gamma_s \quad ?$

Tabla C3-9. Expresiones para la etapa de validación del filtro de Información y del filtro de Información Extendido.

Al igual que ocurría con la no linealidad en la función de transición de estados, los cambios en la etapa de validación no sobrecargarán computacionalmente de forma significativa el algoritmo de inferencia. Además, al igual que en el caso del algoritmo C3-2 la validación se realizará sólo una vez, tan pronto como la nueva medida $\zeta_{s,k}$ sea recibida por el algoritmo, y el resultado de la etapa de validación dependerá únicamente de la información sensorial medida en instantes de muestreo anteriores a k que ya haya sido asimilada por el algoritmo antes de la llegada de la nueva medida.

▪ **Filtro de Información Mixto para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada**

En esta sección se presenta una nueva versión del filtro de Información, capaz de operar con sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordena, cuya función de transición de estados pueda ser lineal o no lineal, y cuyas funciones de medida puedan ser lineales y no lineales. Cuando todas las funciones sean lineales actuará como el filtro de Información del algoritmo C3-2, cuando todas sean no lineales actuará como una versión extendida del mismo, y en el caso en el que haya funciones lineales y no lineales optimizará la sobrecarga computacional del algoritmo realizando las operaciones lineales siempre que éstas sean posibles. El desarrollo del filtro de Información Mixto se basa en el siguiente conjunto de observaciones:

- La etapa de predicción es similar para el caso lineal y no lineal, ya que en ambos casos es necesario pasar la información del estado al espacio de estado y aplicar la función de transición (sea lineal o no) para obtener el nuevo estado del sistema. La diferencia se encuentra en la necesidad de calcular la derivada primera de la función de transición, cuando ésta es no lineal. Para simplificar la exposición del algoritmo C3-3, se utilizará una función auxiliar *ObtenerF*(x, u, t) para obtener el valor de la \mathbf{F}_t , que en el caso lineal devuelve la matriz de transición \mathbf{F}_t y en el caso no lineal devuelve el jacobiano de la función de transición.

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

- En los casos de las etapas de validación y asimilación ocurre algo similar con la matriz $\mathbf{H}_{s,t}$. Para el caso lineal ésta es la matriz de medida, mientras que para el caso no lineal se calcula como el jacobiano de la matriz de medida. Por lo tanto, también se utilizará una función auxiliar $ObtenerH_s(x,t)$ para obtener el valor de la matriz $\mathbf{H}_{s,t}$.
- En un sistema multisensorial se puede dar el caso de que las funciones de medida de algunos sensores sean lineales y las del resto no. Para distinguir los dos casos el algoritmo C3-3 utilizará la variable vectorial booleana S_{Lineal} , que para cada sensor s almacenará si la función de medida es lineal ($S_{Lineal}(s)=true$) o no ($S_{Lineal}(s)=false$). Además, se utilizará la variable matricial booleana $S_{Asimilada}$ para saber si la medida $\zeta_{s,k}$ tomada por el sensor s en el instante k ha sido asimilada y almacenada ($S_{Asimilada}(k,s)=true$) o no ($S_{Asimilada}(k,s)=false$).
- Para los sensores lineales el cálculo de las variables $\mathbf{i}_{s,k}$ (e $\mathbf{I}_{s,k}$) cada vez que se actualiza mediante predicción el valor estimado de la información del estado $\hat{\mathbf{y}}_{k/AASIM \leq k-1}$ (e $\mathbf{Y}_{k/AASIM \leq k-1}$) es una operación costosa e innecesaria. Para evitarla, la información sensorial asociada a los sensores lineales será almacenada en las variables acumulativas de información sensorial lineal $\mathbf{i}L_j$ e $\mathbf{I}L_j$.
- Aunque para los sensores no lineales es necesario recalcular los valores de la información sensorial cuando se asimila una medida temporalmente desordenada, también se utilizarán unas variables acumulativas de información sensorial no lineal $\mathbf{i}NL_k$ e $\mathbf{I}NL_k$. El uso de estas variables aumenta las necesidades de memoria del algoritmo y no es realmente necesario, ya que siempre es posible calcular el valor de las mismas al tener almacenada las medidas los valores de las medidas. Sin embargo, reduce el coste computacional del algoritmo, ya que en la etapa de predicción (2.a) evita el tener que calcular toda la información sensorial no lineal, mientras que en la etapa de asimilación de la medida $\zeta_{s,k}$ evita el tener que calcular la información sensorial no lineal para ese instante k debido al resto de los sensores no lineales.

En el algoritmo C3-3 se presenta la versión del filtro de Información Mixto para sistemas de control con sensores asíncronos que pueden proporcionar las medidas temporalmente desordenadas y con un retardo variable.

1) Inicialización

- Obtener el modelo del sistema (ecuación de transición de estados) y de los sensores (ecuaciones de medida), el vector S_{Lineal} que determinará que sensores s son lineales ($S_{Lineal}(s)=true$) y cuales no ($S_{Lineal}(s)=false$), y la matriz $S_{Asimilada}$, inicializada a false, que determinará si el algoritmo ha asimilado una medida para el sensor s en el instante k ($S_{Asimilada}(k,s)=true$) o no $S_{Asimilada}(k,s)=false$.
- Fijar el tamaño de la ventana de tiempo n .
- $t=1$

- d. Inicializar las variables asociadas al instante $t=1$
- $$\mathbf{F}_0 = \text{ObtenerF}(\hat{\mathbf{x}}_0, \mathbf{u}_0, 0)$$
- $$\mathbf{Y}_{1/AASIM \leq 0} = (\mathbf{F}_0 \mathbf{P}_0 \mathbf{F}_0^T + \mathbf{Q}_0)^{-1} \quad \hat{\mathbf{y}}_{1/AASIM \leq 0} = \mathbf{Y}_{1/AASIM \leq 0} \cdot \mathbf{f}(\hat{\mathbf{x}}_0, \mathbf{u}_0, 0)$$
- $$\mathbf{iL}_1 = \mathbf{0}, \mathbf{IL}_1 = \mathbf{0}, \mathbf{iNL}_1 = \mathbf{0}, \mathbf{INL}_1 = \mathbf{0}$$
- 2) Bucle principal: **while(true)**
- a. Si ha pasado el tiempo de un muestreo, hay que inicializar las variables del nuevo instante de muestreo, y posiblemente eliminar las variables del periodo más antiguo:
- i. $t=t+1$
 - ii. Se asimila la información del instante anterior $t-1$

$$\hat{\mathbf{y}}_{t-1/AASIM \leq t-1} = \hat{\mathbf{y}}_{t-1/AASIM \leq t-2} + \mathbf{iL}_{t-1} + \mathbf{iNL}_{t-1}$$

$$\mathbf{Y}_{t-1/AASIM \leq t-1} = \mathbf{Y}_{t-1/AASIM \leq t-2} + \mathbf{IL}_{t-1} + \mathbf{INL}_{t-1}$$
 - iii. Se inicializan las variables para el instante t

$$\hat{\mathbf{x}}_{t-1/AASIM \leq t-1} = \mathbf{Y}_{t-1/AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t-1/AASIM \leq t-1}$$

$$\mathbf{F}_{t-1} = \text{ObtenerF}(\hat{\mathbf{x}}_{t-1/AASIM \leq t-1}, \mathbf{u}_{t-1}, t-1)$$

$$\mathbf{Y}_{t/AASIM \leq t-1} = (\mathbf{F}_{t-1} \mathbf{Y}_{t-1/AASIM \leq t-1}^{-1} \mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})^{-1}$$

$$\hat{\mathbf{y}}_{t/AASIM \leq t-1} = \mathbf{Y}_{t/AASIM \leq t-1} \cdot \mathbf{f}(\hat{\mathbf{x}}_{t-1/AASIM \leq t-1}, \mathbf{u}_{t-1}, t-1)$$

$$\mathbf{iL}_t = \mathbf{0}, \mathbf{IL}_t = \mathbf{0}, \mathbf{iNL}_t = \mathbf{0}, \mathbf{INL}_t = \mathbf{0}$$
 - iv. Si $t-n > 0$ hay que eliminar los elementos asociados al instante $t-n$, es decir las variables:
$$\mathbf{Y}_{t-n/AASIM \leq t-n-1}, \hat{\mathbf{y}}_{t-n/AASIM \leq t-n-1}, \mathbf{u}_{t-n}, \mathbf{iL}_{t-n}, \mathbf{IL}_{t-n}, \mathbf{iNL}_{t-n}, \mathbf{INL}_{t-n}$$

$$\zeta_{s,t-n} \text{ para los } s \text{ no lineales}$$
- b. Si hay una medida $\zeta_{s,k}$ disponible, hay que comprobar si ésta se ha realizado en un k que esta dentro de la ventana temporal n elegida.
- i. Si no ($k > t-n$) la medida no se puede asimilar, por lo que es conveniente mandar un mensaje al usuario para que detecte que con el n elegido se pierden medidas
 - ii. Si ($k > t-n$) la medida se podrá asimilar, siempre que sea válida:
 1. Obtener el valor estimado de la medida
$$\hat{\mathbf{x}}_{k/AASIM \leq k-1} = \mathbf{Y}_{k/AASIM \leq k-1}^{-1} \hat{\mathbf{y}}_{k/AASIM \leq k-1}$$

$$\mathbf{H}_{s,k} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k)$$

$$\hat{\mathbf{z}}_{s,k/AASIM \leq k-1} = h_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k)$$

$$\mathbf{S}_{s,k/AASIM \leq k-1} = \mathbf{H}_{s,k} \mathbf{Y}_{k/AASIM \leq k-1}^{-1} \mathbf{H}_{s,k}^T + \mathbf{R}_{s,k}$$
 2. Realizar el test chi-cuadrado de validación basado en el valor estimado de la medida
$$\hat{\mathbf{r}}_{s,k/AASIM \leq k-1} = \zeta_{s,k} - \hat{\mathbf{z}}_{s,k/AASIM \leq k-1}$$

$$\text{Test: } \hat{\mathbf{r}}_{s,k/AASIM \leq k-1}^T \mathbf{S}_{s,k/AASIM \leq k-1}^{-1} \hat{\mathbf{r}}_{s,k/AASIM \leq k-1} \leq \gamma_s \quad ?$$
 3. Si la medida es válida:
 - a. Obtener la información sensorial asociada a la medida (en b.ii.1 ya se ha obtenido la $\mathbf{H}_{s,k}$ y $\hat{\mathbf{x}}_{k/AASIM \leq k-1}$) e indicar que va a ser asimilarada

C3.2. *Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada*

```

if SLineal(s) = true
     $\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \boldsymbol{\zeta}_{s,k}$ 
     $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}$ 
else
     $\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} (\boldsymbol{\zeta}_{s,k} - h_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k) + \mathbf{H}_{s,k} \hat{\mathbf{x}}_{t/AASIM \leq k-1})$ 
     $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}$ 
    Almacenar  $\boldsymbol{\zeta}_{s,k}$ 
end
SAsimilada(k, s) = true

b. Añadir la nueva información sensorial al resto de la
información sensorial, del mismo tipo, asociada a
dicho periodo de muestreo:
if SLineal(s) = 1
     $\mathbf{iL}_k = \mathbf{iL}_k + \mathbf{i}_{s,k}$  ,  $\mathbf{IL}_k = \mathbf{IL}_k + \mathbf{I}_{s,k}$ 
else
     $\mathbf{iNL}_k = \mathbf{iNL}_k + \mathbf{i}_{s,k}$  ,  $\mathbf{INL}_k = \mathbf{INL}_k + \mathbf{I}_{s,k}$ 
end

c. Si  $k \neq t$ , hay que propagar la nueva información
sensorial hacia delante, de forma que se actualicen el
término aditivo de la etapa de asimilación que incluye
la información sobre el estado en instante de muestreo
anteriores:
Se asimila la información del instante k y se realiza
la primera predicción
 $\hat{\mathbf{y}}_{k/AASIM \leq k} = \hat{\mathbf{y}}_{k/AASIM \leq k-1} + \mathbf{iL}_k + \mathbf{iNL}_k$ 
 $\mathbf{Y}_{k/AASIM \leq k} = \mathbf{Y}_{k/AASIM \leq k-1} + \mathbf{IL}_k + \mathbf{INL}_k$ 
 $\hat{\mathbf{x}}_{k/AASIM \leq k} = \mathbf{Y}_{k/AASIM \leq k}^{-1} \hat{\mathbf{y}}_{k/AASIM \leq k}$ 
 $\mathbf{F}_k = \text{ObtenerF}(\hat{\mathbf{x}}_{k/AASIM \leq k}, \mathbf{u}_k, k)$ 
 $\mathbf{Y}_{k+1/AASIM \leq k} = (\mathbf{F}_k \mathbf{Y}_{k/AASIM \leq k}^{-1} \mathbf{F}_k^T + \mathbf{Q}_k)^{-1}$ 
 $\hat{\mathbf{y}}_{k+1/AASIM \leq k} = \mathbf{Y}_{k+1/AASIM \leq k} \cdot f(\hat{\mathbf{x}}_{k/AASIM \leq k}, \mathbf{u}_k, k)$ 
for  $j = k+1:t-1$ 
    Se recalcula la información sensorial no lineal
     $\hat{\mathbf{x}}_{j/AASIM \leq j-1} = \mathbf{Y}_{j/AASIM \leq j-1}^{-1} \hat{\mathbf{y}}_{j/AASIM \leq j-1}$ 
     $\mathbf{iNL}_j = \mathbf{0}$ ,  $\mathbf{INL}_j = \mathbf{0}$ 
    for  $s = 1:S$ 
        if SLineal(s) = false  $\wedge$  SAsimilada(j, s) = true
             $\mathbf{H}_{s,j} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{j/AASIM \leq j-1}, j)$ 
             $\mathbf{i}_{s,j} = \mathbf{H}_{s,j}^T \mathbf{R}_{s,j}^{-1} (\boldsymbol{\zeta}_{s,j} - h_s(\hat{\mathbf{x}}_{j/AASIM \leq j-1}, j) + \mathbf{H}_{s,j} \hat{\mathbf{x}}_{j/AASIM \leq j-1})$ 
             $\mathbf{I}_{s,j} = \mathbf{H}_{s,j}^T \mathbf{R}_{s,j}^{-1} \mathbf{H}_{s,j}$ 
             $\mathbf{iNL}_j = \mathbf{iNL}_j + \mathbf{i}_{s,j}$ 
             $\mathbf{INL}_j = \mathbf{INL}_j + \mathbf{I}_{s,j}$ 
        end
    end
end
end

```

```

Se asimila la información sensorial para el
instante j y se realiza la siguiente predicción
 $\hat{\mathbf{y}}_{j|AASIM \leq j} = \hat{\mathbf{y}}_{j|AASIM \leq j-1} + \mathbf{i}L_j + \mathbf{i}NL_j$ 
 $\mathbf{Y}_{j|AASIM \leq j} = \mathbf{Y}_{j|AASIM \leq j-1} + \mathbf{I}L_j + \mathbf{I}NL_j$   $\hat{\mathbf{x}}_{j|AASIM \leq j} = \mathbf{Y}_{j|AASIM \leq j}^{-1} \hat{\mathbf{y}}_{j|AASIM \leq j}$ 
 $\mathbf{F}_j = \text{ObtenerF}(\hat{\mathbf{x}}_{j|AASIM \leq j}, \mathbf{u}_j, j)$ 
 $\mathbf{Y}_{j+1|AASIM \leq j} = (\mathbf{F}_j \mathbf{Y}_{j|AASIM \leq j}^{-1} \mathbf{F}_j^T + \mathbf{Q}_j)^{-1}$ 
 $\hat{\mathbf{y}}_{j+1|AASIM \leq j} = \mathbf{Y}_{j+1|AASIM \leq j} \cdot f(\hat{\mathbf{x}}_{j|AASIM \leq j}, \mathbf{u}_j, j)$ 
end
Se recalcula la información sensorial no lineal del
último instante de muestreo
 $\hat{\mathbf{x}}_{t|AASIM \leq t-1} = \mathbf{Y}_{t|AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t|AASIM \leq t-1}$ 
 $\mathbf{i}NL_t = \mathbf{0}, \mathbf{I}NL_t = \mathbf{0}$ 
for s = 1 : S
    if SLineal(s) = false  $\wedge$  SAsimilda(t, s) = true
         $\mathbf{H}_{s,t} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{t|AASIM \leq t-1}, t)$ 
 $\mathbf{i}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} (\zeta_{s,t} - h_s(\hat{\mathbf{x}}_{t|AASIM \leq t-1}, t) + \mathbf{H}_{s,t} \hat{\mathbf{x}}_{t|AASIM \leq t-1})$ 
 $\mathbf{I}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t}$ 
 $\mathbf{i}NL_t = \mathbf{i}NL_t + \mathbf{i}_{s,t}$ 
 $\mathbf{I}NL_t = \mathbf{I}NL_t + \mathbf{I}_{s,t}$ 
    end
end
end

```

Algoritmo C3-3. Filtro de Información Mixto para un sistema de control con múltiples sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada, y cuya función de transición puede ser lineal o no lineal, y cada una de sus funciones de medida lineales o no lineales.

En el capítulo siguiente se realiza un análisis comparativo del funcionamiento del filtro para tres sistemas ejemplo diferente. Durante el mismo se observará la penalización computacional introducida por el algoritmo, respecto al caso puramente lineal, cuando la función de transición es no lineal y cuando las funciones de medida de los sensores utilizados son no lineales. También se comprobará la importancia del término corrector de la medida en la etapa de asimilación de medidas provenientes de sensores no lineales y se analizará la necesidad del recálculo de la información sensorial asociada a los sensores no lineales cuando el algoritmo recibe medidas retardadas provenientes de otros sensores.

C3.2.3. Características relevantes del filtro de Información Mixto para estimar el estado de un sistema de control con medidas temporalmente desordenadas y análisis comparativo con otros algoritmos de fusión multisensorial similares

El algoritmo de fusión multisensorial C3-3, desarrollado para estimar el estado de un sistema de control con múltiples sensores asíncronos que pueden proporcionar sus medidas al sistema de fusión de forma temporalmente desordenada, basa su funcionamiento en las ecuaciones de dos versiones diferentes del filtro de Información: el filtro de Información para sistemas multisensoriales lineales y el filtro de Información Extendido para sistemas multisensoriales no lineales.

La elección del filtro de Información como soporte del algoritmo es una consecuencia directa del análisis exhaustivo, realizado en el capítulo anterior, de las operaciones llevadas a cabo por el algoritmo de inferencia exacto para redes bayesianas con variables gaussianas continuas, cuando éste es aplicado para estimar el estado de un sistema lineal de control multisensorial cuyo comportamiento es modelado mediante una red bayesiana. Los resultados de dicho análisis han sido un conjunto de algoritmos de inferencia exactos sobre el árbol de unión asociado a la red bayesiana, que son capaces de tratar automáticamente la información sensorial temporalmente retardada y desordenada, y se caracterizan por ser computacionalmente equivalentes al filtro de Información cuando la información sensorial es recibida por los algoritmos de forma temporalmente ordenada. Sin embargo, para este caso particular, la complejidad operacional y las necesidades de memoria de los algoritmos de inferencia exactos son significativamente mayores que los del filtro de Información, por lo que se decidió sustituir el último algoritmo exacto desarrollado, el C3-1, por una nueva versión del filtro de Información que tratase la información temporalmente desordenada de forma semejante a la seguida por los algoritmos de inferencia exacta.

La nueva versión del filtro de Información para fusión multisensorial sobre sistemas puramente lineales, recogida en algoritmo C3-2, es una solución que para cada instante de muestreo \mathbf{k} almacena en variables diferentes:

- la información disponible sobre el estado del sistema $\hat{\mathbf{y}}_{k/AASIM \leq k-1}$ (y su covarianza $\mathbf{Y}_{k/AASIM \leq k-1}$) que ha sido propagada, mediante una etapa de predicción, desde el instante de muestreo anterior $\mathbf{k}-1$,
- toda la información sensorial \mathbf{i}_k (y su covarianza \mathbf{I}_k) recibida por el algoritmo y correspondiente a dicho instante \mathbf{t} .

Debido a esta separación, cuando la medida $\zeta_{s,k}$, tomada por el sensor s en el instante k , es recibida por el algoritmo C3-2 en un instante de muestreo t posterior ($t > k$), es posible ir propagando hasta el instante t la información sensorial debida a la nueva medida, a la vez que se repropaga hasta el mismo instante t la información sensorial recibida por el algoritmo anteriormente y que ya se encuentra asociada a los instantes de muestreo posteriores. Es decir, el algoritmo C3-2 retrasa simultáneamente la nueva información sensorial y la información sensorial asociada a instantes de muestreo posteriores a la última información recibida que ya ha sido asimilada anteriormente por el algoritmo.

La repropagación de la información sensorial no es una solución novedosa, aunque ésta suele ser rechazada debido al gasto computacional y de memoria que conlleva. Sin embargo, en la solución presentada en el algoritmo C3-2 para el caso lineal multisensorial, se ahorra memoria ya que todas las medidas para un instante de muestreo pueden ser acumuladas como información sensorial en dos variables (\mathbf{i}_k para la información sensorial e \mathbf{I}_k para su covarianza) en vez de en una variable diferente para cada medida e instante de muestreo. Además, al acumular las medidas en forma de información sensorial ($\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \zeta_{s,k}$ e $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}$) el principal gasto computacional asociado a la asimilación de la medida se elimina, y únicamente es necesario realizar predicciones sucesivas para repropagar la información sensorial hasta el instante de muestreo t . Estas predicciones también son necesarias para propagar la información sensorial debida a la nueva medida, por lo que el gasto computacional extra debido a la repropagación de la información sensorial anteriormente asimilada puede ser considerado despreciable.

Este ahorro computacional y de memoria es la consecuencia directa de la elección del filtro de Información como base del algoritmo de fusión multisensorial frente a su equivalente en el espacio de estados, el filtro de Kalman. Las expresiones del filtro de Kalman multisensorial lineal (que se encuentran recogidas en la tabla A1.8 del apéndice I) no permitirían dicho ahorro, porque el término de la asimilación depende de la diferencia entre la medida y el valor estimado de la misma a partir del valor estimado del estado. Por lo tanto, una repropagación similar asociada a la asimilación de una medida temporalmente retardada conllevaría el cálculo de todas las expresiones de la etapa de asimilación, además de la necesidad de almacenar en variables diferentes todas las medidas ya asimiladas, con objeto de poder realizar al recálculo de la información sensorial.

La versión del filtro de Información Mixto C3-3 extiende el ámbito del uso del algoritmo C3-2 al caso de sistema no lineales o sistemas mixtos, con linealidad en unas expresiones y no linealidad en otras.

Cuando la no linealidad únicamente se da en la ecuación de transición de estados, las propiedades del algoritmo C3-3 son similares a las propiedades del algoritmo C3-2, con la salvedad de que para

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

el caso no lineal surge la necesidad de tener que recalcular la derivada primera de la función de transición $f(x,u,t)$ cada vez que se realiza una predicción sobre un instante de muestreo. Esta sobrecarga computacional es necesaria tanto para realizar la propagación de la nueva información sensorial como para llevar a cabo la repropagación de la información sensorial asimilada, por lo que al igual que en el caso puramente lineal, el gasto asociado a la repropagación puede ser considerado despreciable. Además, se pueden seguir utilizando las variables de información sensorial, identificadas como $\mathbf{i}L_k$ e $\mathbf{I}L_k$ en el algoritmo C3-3, para almacenar toda la información asociada al instante de muestreo \mathbf{k} que ya ha sido asimilada por el algoritmo. Por lo tanto, el ahorro computacional y de memoria asociado a la asimilación de la información sensorial se mantiene.

En los casos en que las ecuaciones de medida son no lineales el ahorro computacional y de memoria asociado a la etapa de asimilación desaparece, ya que para estos sensores la información sensorial $\mathbf{i}_{s,j} = \mathbf{H}_{s,j}^T \mathbf{R}_{s,j}^{-1} (\zeta_{s,j} - h_s(\hat{\mathbf{x}}_{j|AASIM \leq j-1}, j) + \mathbf{H}_{s,j} \hat{\mathbf{x}}_{j|AASIM \leq j-1})$ y su covarianza $\mathbf{I}_{s,j} = \mathbf{H}_{s,j}^T \mathbf{R}_{s,j}^{-1} \mathbf{H}_{s,j}$ depende del estado del sistema, y éste valor cambia durante la asimilación de información sensorial retardada. Por lo tanto, para los sensores no lineales será necesario almacenar los valores de las medidas ya asimiladas para todos los instantes de muestreo dentro de la ventana temporal elegida, y recalcular la información sensorial $\mathbf{i}_{s,j}$ (e $\mathbf{I}_{s,j}$) asociada a dichas medidas cada vez que el sistema asimila una medida tomada por cualquier sensor en un instante de muestreo anterior. Sin embargo, como este cálculo no es necesario para los sensores lineales, en los sistemas con sensores lineales y no lineales la información asociada a los sensores lineales se almacenará en las variables de información sensorial lineal $\mathbf{i}L_k$ e $\mathbf{I}L_k$, con el objeto de minimizar el gasto computacional y las necesidades de memoria asociados a la etapa de asimilación. El uso de las variables de información no lineal $\mathbf{i}NL_k$ e $\mathbf{I}NL_k$ es opcional, ya que su uso enfrenta una reducción computacional pequeña ante un aumento considerable de las necesidades de memoria.

Por último, destacar que el algoritmo de fusión multisensorial C3-3 incluye una etapa opcional de validación de medidas, basada en el test chi-cuadrado, que se realiza únicamente en el instante en el que la medida llega al algoritmo de fusión multisensorial, y cuyo resultado será mantenido durante todas las etapas del algoritmo.

Además de los métodos propuestos en este capítulo, durante los últimos años se han desarrollado otros métodos que resuelven el problema de la asimilación de medidas temporalmente desordenadas para sistemas de control siguiendo varias filosofías diferentes. En los siguientes apartados se analizan las características de estos métodos y se comparan con los algoritmos C3-2 y C3-3 de este capítulo.

▪ **Otros métodos para la asimilación asíncrona de medidas temporalmente desordenadas**

El creciente interés del problema del tratamiento de las medidas realizadas por sensores asíncronos cuyo funcionamiento no asegura que la información sensorial este disponible de forma temporalmente ordenada ha provocado el desarrollo reciente de diferentes algoritmos de asimilación asíncrona de medidas temporalmente desordenadas. Una solución menos novedosa puede encontrarse en [Thomopoulos1988].

En el análisis comparativo realizado en este apartado nos centraremos únicamente en los recientes métodos de filtrado que permiten estimar el estado del sistema en el último instante de muestreo a partir de toda la información sensorial disponible hasta dicho instante y que soportan la asimilación de medidas de forma temporalmente desordenada.

Métodos basados en el filtro de Kalman y la retrodicción del estado y su covarianza (sistemas lineales)

La primera de las filosofías para resolver el problema de la asimilación de una medida $\zeta_{s,\tau}$ temporalmente retardada respecto al último instante de muestreo \mathbf{t} , se basa en la retrodicción (predicción hacia atrás) del estado del sistema hasta el instante de medida τ . Esta operación se fundamenta en la siguiente expresión de propagación inversa que, partiendo de la función de transición de estados entre los instantes τ y \mathbf{t} , permite obtener el valor del estado en el instante τ a partir del valor en el instante posterior \mathbf{t} y del ruido del sistema $v_{t \rightarrow \tau}$

$$\mathbf{x}_t = \mathbf{F}_{t \rightarrow \tau} \mathbf{x}_\tau + v_{t \rightarrow \tau} \rightarrow \mathbf{x}_\tau = \mathbf{F}_{t \rightarrow \tau}^{-1} \cdot (\mathbf{x}_t - v_{t \rightarrow \tau}) \quad (\text{C3.1})$$

Es importante destacar que estos métodos suponen que la matriz de transición de estados \mathbf{F} es invertible y función del tiempo transcurrido entre los instantes τ y \mathbf{t} , suposiciones siempre válidas para los sistemas discretos que se derivan de la discretización de sistemas lineales continuos [Gelb1974].

A partir de dicha expresión, en [Bar-Shalom2002a] se desarrollan un conjunto de ecuaciones que permiten asimilar de forma exacta una medida retardada dentro de un único periodo de muestro $(\mathbf{t}-1 < \tau < \mathbf{t})$. El método se puede dividir en dos etapas diferenciadas: en la primera se retrodice el valor estimado del estado del sistema y su covarianza hasta el instante τ , y en el segundo se asimila la medida $\zeta_{s,\tau}$. Las expresiones de asimilación se realizan en el espacio de estados, por lo que mantienen una cierta analogía con las expresiones de asimilación habituales del filtro de Kalman para un sistema monosensorial.

C3.2. *Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada*

Para retardos superiores a un periodo de muestreo ($\mathbf{t-r-1} < \tau < \mathbf{t-r}$), en [Bar-Shalom2002a] se propone una alternativa que permite retrodecir el valor estimado del estado del sistema hasta el instante de medida τ , mediante la combinación una etapa de alisado no convencional hasta el instante de muestreo posterior a la medida $\mathbf{t-r}$ (etapa para la que es necesario almacenar la innovación o las medidas desde \mathbf{t} hasta $\mathbf{t-r}$) y una etapa de retrodicción desde $\mathbf{t-r}$ hasta τ , igual a la propuesta en el caso de que la medida se encuentre retardada dentro de un único periodo.

En [Bar-Shalom2002b] se soluciona el problema de la asimilación de medidas con retardos superiores a un periodo de muestreo ($\mathbf{t-r-1} < \tau < \mathbf{t-r}$) por medio de una única etapa de retrodicción. El método se fundamenta en la definición de una medida equivalente $\mathbf{Z}_{t-r:t}$ que en el instante \mathbf{t} engloba a todas las medias hasta el instante $\mathbf{t-r}$ $\mathbf{Z}_{t-r:t} = \{\mathbf{z}_{t-r}, \mathbf{z}_{t-r+1}, \dots, \mathbf{z}_t\}$, y la reformulación para la medida equivalente $\mathbf{Z}_{t-r:t}$ del algoritmo presentado en [Bar-Shalom2002a] para las medidas retardadas un único instante de muestreo. Durante la reformulación se realiza una aproximación, por lo que el algoritmo presentado en [Bar-Shalom2002b] es un método subóptimo para estimar el estado de un sistema con medidas temporalmente desordenadas. Además, para el caso en el que el retardo \mathbf{r} es variable, es necesario almacenar los valores estimados del estado del sistema $\hat{\mathbf{x}}_{t|t-r}$ y su covarianza $\mathbf{P}_{t|t-r}$ durante una ventana de tiempo \mathbf{n} que asegure que todas las medidas retardadas puedan ser asimiladas.

Por último, es importante destacar que los métodos basados en la retrodicción del estado propuestos en [Bar-Shalom2002a] y [Bar-Shalom2002b] no sirven para resolver el problema multisensorial de forma óptima cuando el retardo es mayor que un periodo de muestreo. Además, la formulación propuesta en los dos artículos no considera la existencia de una señal de control en la función de transición de estados, y no se analiza la utilización de una etapa de validación de medidas.

Además de estos dos métodos, en [Bar-Shalom2002a], [Hilton1993] y [Mallick2001] se proponen otros métodos subóptimos basados en la retrodicción del estado, en los que se supone que la retrodicción del ruido del sistema $\mathbf{v}_{t \rightarrow \tau}$ es nula.

Métodos basados en el filtro de Kalman y el precálculo de un conjunto de valores que permita la asimilación de la medida tan pronto como ésta se encuentre disponible (sistemas lineales)

Esta segunda filosofía basa su funcionamiento en el filtro de Kalman para el caso monosensorial y en un conjunto de ecuaciones que permiten la asimilación directa de la medida ζ_k realizada en el instante \mathbf{k} sobre el valor estimado por el filtro de Kalman habitual en el instante \mathbf{t} . Las expresiones utilizadas para la asimilación de medidas retardadas, recogidas en las ecuaciones (C3-2) y (C3-3), tienen una forma similar a las expresiones de asimilación habitual (presentadas en la tabla A1.3 del

apéndice I) pero las matrices \mathbf{W} y \mathbf{S} , y el vector de estado utilizado para estimar el valor de medida $\hat{\mathbf{x}}_{k/t}$ se obtienen de forma diferente, por medio de un conjunto de operaciones que son ejecutadas desde el instante \mathbf{k} , cíclicamente y a la vez que se realizan las etapas habituales del filtro de Kalman.

$$\hat{\mathbf{x}}_{t/t,k} = \hat{\mathbf{x}}_{t/t} + \mathbf{W}_{k,t}(\zeta_k - \mathbf{H}_k \hat{\mathbf{x}}_{k/t}) \quad (\text{C3.2})$$

$$\mathbf{P}_{t/t,k} = \mathbf{P}_{t/t} - \mathbf{W}_{k,t} \mathbf{S}_{k,t} \mathbf{W}_{k,t}^T \quad (\text{C3.3})$$

El método óptimo propuesto en [ZhangK2002], denominado algoritmo I, que tiene como objetivo principal minimizar el número de variables que hay que almacenar para asimilar la medida retardada ζ_k , obtiene el los valores de la tripleta $(\mathbf{W}_{k,t}, \mathbf{S}_{k,t}, \hat{\mathbf{x}}_{k/t})$ justo antes de realizar la asimilación de la medida retardada de un conjunto $\Omega_{k,t}$ más amplio de variables auxiliares. Los valores de las variables del conjunto $\Omega_{k,j} = \{\hat{\mathbf{x}}_{k/j}, \mathbf{P}_{k/j}, \mathbf{T}_j, \mathbf{W}_j\}$ para el instante \mathbf{j} posterior a \mathbf{k} son calculados de forma iterativa (utilizando un conjunto de ecuaciones que se pueden encontrar en la literatura propuesta) a la vez que se obtiene, sin utilizar el valor de la medida no disponible ζ_k , el valor estimado del estado $\hat{\mathbf{x}}_{j/j}$ (y su covarianza $\mathbf{P}_{j/j}$) por medio de las ecuaciones de predicción y asimilación habituales del filtro de Kalman.

En el mejor de los casos para el algoritmo óptimo propuesto, que es aquel en el que se sabe el instante de muestreo \mathbf{k} en el que falta la media ζ_k , el cálculo del conjunto de variables auxiliares $\Omega_{k,k}$ comenzará en el instante anterior \mathbf{k} . A partir de dicho instante, y hasta que la medida ζ_k haya sido recibida por el algoritmo en el instante \mathbf{t} , el algoritmo almacenará y actualizará el conjunto de variables $((\hat{\mathbf{x}}_{j/j}, \mathbf{P}_{j/j}) \cup \Omega_{k,j})$. De esta forma, las dos primeras variables se utilizarán para estimar de la forma habitual el estado del sistema y su covarianza, y el resto para poder asimilar (mediante las expresiones (C3-2), (C3-3), y aquellas que permiten obtener de las variables del conjunto $\Omega_{k,t}$ los valores de la tripleta $(\mathbf{W}_{k,t}, \mathbf{S}_{k,t}, \hat{\mathbf{x}}_{k/t})$) la medida ζ_k tan pronto como ésta esté disponible.

Sin embargo, en el caso general en el que se desconozca el instante exacto \mathbf{k} en el que falta una medida, hecho que se da cuando faltan las medidas asociadas a distintos periodos de muestreo, el conjunto de variables Ω^j que el algoritmo almacena y actualiza crece para cada instante de muestreo \mathbf{j} con un nuevo subconjunto $(\Omega^j = \bar{\Omega}^{j-1} \cup \{\hat{\mathbf{x}}_{j/j}, \mathbf{P}_{j/j}, \hat{\mathbf{x}}_{k/j}, \mathbf{P}_{k/j}, \mathbf{T}_j, \mathbf{W}_j\})$. De esta forma se asegura que el conjunto $\Omega_{k,j}$ asociado a la medida retardada ζ_k estará disponible para poder realizar la asimilación retardada. El creciente tamaño del conjunto de variables almacenadas, y cuyo valor hay que actualizar en cada instante de muestreo, se puede limitar, de forma análoga a la que se realizó en los algoritmos C3-2 y C3-3, mediante el uso de una ventana temporal de tamaño \mathbf{n} .

Además de este método, en [ZhangK2002] se presenta un método subóptimo, denominado algoritmo II, que permite reducir el número de variables almacenadas en cada periodo de muestreo a 4, que se van actualizando en todos los periodos de muestreo sucesivos, cuando se considera que

C3.2. Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada

el ruido del sistema $v_{t \rightarrow \tau}$ (presentado en la ecuación (C3-1)) es nulo. Este método subóptimo, cuyo funcionamiento es comparado con el del filtro de Partículas en [Mallick2003], es un caso general de los métodos subóptimos en [Bar-Shalom2002a] y [Hilton1993], ya que los métodos de [ZhangK2002] evitan la suposición de la invertibilidad de la matriz de transición de estados \mathbf{F} a costa de precalcular cíclicamente un conjunto de variables, y por lo tanto, pueden ser utilizados en sistemas lineales discretos que no se han obtenido como la discretización de sistemas lineales continuos.

Por último es importante destacar que en la formulación de los métodos de [ZhangK2002] no se considera la existencia de una señal de control en la función de transición de estados, y no se analiza la utilización de una etapa de validación de medidas. Además, la formulación de la asimilación retardada se realiza en el espacio de estados con una fórmula equivalente a la del filtro de Kalman monosensorial, que no tiene en cuenta las modificaciones necesarias para la asimilación de las medidas en sistemas multisensoriales. Por lo tanto, estos métodos tendrán un comportamiento subóptimo para el caso de sistemas multisensoriales.

Método basado en el filtro de Información y el almacenamiento mínimo de variables (sistemas lineales)

En [Nettleton2001] se presenta un método para asimilar medidas retardadas y temporalmente desordenadas fundamentado en el filtro de Información en vez de en el filtro de Kalman. Dicho método presupone, al igual que los métodos basados en la retrodicción del estado, la invertibilidad de la matriz de transición de estados \mathbf{F}_t con el objeto de poder realizar la etapa de propagación del filtro de Información directamente en el espacio de información, y de esta forma evitar la doble conversión de espacios que se realiza en dicha etapa en el algoritmo de Información para sistemas lineales de control C3-2.

La metodología seguida en [Nettleton2001] es similar a la de nuestro algoritmo: se almacena un conjunto de valores y cuando la medida temporalmente retardada llega al algoritmo esta es asimilada con la información del estado asociada a su instante de medida y propagada hacia delante. Las diferencias se encuentran en el número de variables almacenadas y en la forma de propagar la información sensorial retardada, y son consecuencia directa de la realización de la etapa de propagación en el espacio de información:

- El algoritmo propuesto en [Nettleton2001] únicamente almacena dos variables por instante de muestreo \mathbf{j} , toda la información del estado para el instante de muestreo (que según la notación del algoritmo C3-2 es $\hat{\mathbf{y}}_{j|AASIM \leq j}$) y una variable $\mathbf{M}_{j|AASIM \leq j}$ relacionada, a través de la inversa de la matriz de transición, con la covarianza asociada a la información de dicho estado ($\mathbf{M}_{j|AASIM \leq j} = \mathbf{F}_j^{-T} \mathbf{Y}_{j|AASIM \leq j} \mathbf{F}_j^{-1}$). El algoritmo C3-2 almacena para cada instante de

muestreo j cuatro variables, dos relacionadas con la información sensorial asociada a dicho instante (\mathbf{i}_j e \mathbf{I}_j) y otras dos con la información sensorial que ha sido propagada desde instantes anteriores de muestro ($\hat{\mathbf{y}}_{j|AASIM \leq j-1}$ e $\mathbf{Y}_{j|AASIM \leq j-1}$). En ambos casos, además es necesario almacenar también el valor de la señal de control \mathbf{u}_j .

- El algoritmo en [Nettleton2001] propaga hacia delante únicamente la información sensorial $\mathbf{i}_{s,j}^*$ asociada a la medida retardada $\zeta_{s,k}$, y la asimila a toda la información que se tiene sobre el estado en cada instante $\hat{\mathbf{y}}_{j|AASIM \leq j}$, mientras que el algoritmo C3-2 repropaga $\hat{\mathbf{y}}_{j|AASIM \leq j}$ (que almacena simultáneamente la información asociada a la nueva medida y la información sobre el estado para cada instante), y la asimila a la información sensorial almacenada por el algoritmo \mathbf{i}_j . La complejidad operacional asociada a la predicción de la información sensorial $\mathbf{i}_{s,j}^*$ (que se recoge en las expresiones (13) y (14) del algoritmo presentado en [Nettleton2001]) es superior a la complejidad operacional asociada a la etapa de propagación habitual del filtro de Información, a pesar de que en este último caso sea necesario realizar una doble conversión del espacio de trabajo.

Por lo tanto, el algoritmo de [Nettleton2001] ahorra memoria a costa de aumentar la complejidad operacional asociada a la asimilación de medidas retardadas, y de suponer la invertibilidad de la matriz de transición de estados \mathbf{F}_t . En su formulación se considera la existencia de la señal de control en la matriz de transición de estados, y debido a realizar la asimilación en el espacio de información este algoritmo también resulta óptimo para el caso multisensorial.

Aunque la existencia de la etapa de validación, basada en la distancia entre el valor estimado y real de la medida, no es considerada en el algoritmo de [Nettleton2001], ésta puede ser fácilmente incluida ya que se almacena para cada instante de muestreo información sobre el estado y una variable relacionada con su covarianza. Sin embargo, es importante tener en cuenta que las variables almacenadas en dicho artículo corresponden a $\hat{\mathbf{y}}_{j|AASIM \leq j}$ y $\mathbf{M}_{j|AASIM \leq j} = \mathbf{F}_j^{-T} \mathbf{Y}_{j|AASIM \leq j} \mathbf{F}_j^{-1}$, mientras que en el algoritmo C3-2 se valida sobre las variables $\hat{\mathbf{y}}_{j|AASIM \leq j-1}$ e $\mathbf{Y}_{j|AASIM \leq j-1}$, por lo que la validación realizada será diferente. En el primer caso, la información sensorial asociada al instante de muestreo \mathbf{k} de los sensores más rápidos será utilizada para validar la información sensorial asociada al mismo instante de los sensores más lentos, mientras que en el segundo (al almacenar el algoritmo de forma separada la información del estado relacionada con las medidas anteriores a \mathbf{k} y la información sensorial asociada a dicho instante), no se utilizará la información sensorial asociada a un mismo instante \mathbf{k} para validar la información de los otros sensores.

Métodos basados en el filtro de Kalman y el almacenamiento mínimo de variables (sistemas lineales)

A medio camino entre los métodos propuestos en [ZhangK2002] y [Nettleton2001] se encuentran los algoritmos propuestos en [ZhangK2003]. Por una parte, se utilizan las expresiones de asimilación retardada (C3-2) y (C3-3), dependientes de la tripleta $(\mathbf{W}_{k,t}, \mathbf{S}_{k,t}, \hat{\mathbf{x}}_{k|t})$, para asimilar la medida retardada $\zeta_{s,k}$, mientras que por la otra, se pospone el cálculo de los valores de la tripleta hasta el instante en el que el algoritmo recibe la medida retardada, con el objeto de minimizar el número de variables que hay que almacenar para cada periodo de muestreo. Para cada uno de los algoritmos de [ZhangK2002] se propone una versión en [ZhangK2003]: el nuevo algoritmo óptimo I únicamente necesita almacenar las variables $\hat{\mathbf{x}}_{j|AASIM \leq j}$ e $\mathbf{P}_{j|AASIM \leq j}$, mientras que el nuevo algoritmo subóptimo II únicamente necesita almacenar la variable $\mathbf{P}_{j|AASIM \leq j}$. Una vez que la medida está disponible, las variables almacenadas son utilizadas para obtener los valores de $(\mathbf{W}_{k,t}, \mathbf{S}_{k,t}, \hat{\mathbf{x}}_{k|t})$, y actualizadas para el caso en el que se reciba otra medida retardada.

Las variables de la versión óptima y las del algoritmo propuesto en [Nettleton2001] son equivalentes, aunque cada una se encuentra en un espacio de trabajo diferente: las primeras en el espacio de estados y las segundas en el espacio de información sobre el estado. De ahí la semejanza existente entre los dos métodos. Sin embargo, existen tres diferencias fundamentales, ya que en el algoritmo I: (1) no se supone la invertibilidad de la matriz \mathbf{F}_t , (2) no se ha contemplado la existencia de la señal de control durante su formulación, y (3) su funcionamiento será subóptimo en el caso multisensorial al realizar la asimilación en el espacio de estados con las expresiones de asimilación retardada (C3-2) y (C3-3).

Aunque la existencia de la etapa de validación, basada en la distancia entre el valor estimado y real de la medida, no es tampoco considerada en los algoritmos propuestos en [ZhangK2003], ésta puede ser fácilmente incluida al igual que en el algoritmo de [Nettleton2001], debido a que se almacenan las variables $\hat{\mathbf{x}}_{j|AASIM \leq j}$ e $\mathbf{P}_{j|AASIM \leq j}$. Las diferencias entre la etapa de validación para las variables almacenadas en [ZhangK2003] y el algoritmo C3-2 de este capítulo han sido ya expuestas en la subsección anterior.

Métodos basados en el filtro de Kalman y la reformulación del problema por medio del uso de una variable de estado aumentada (sistemas lineales)

Una alternativa a los métodos anteriormente expuestos, basada en un análisis bayesiano del problema, se encuentra en reformular el problema por medio del uso de una variable de estado aumentada \mathbf{X} que tiene en cuenta el estado los últimos n instantes ($\mathbf{X} = [\mathbf{x}_k; \mathbf{x}_{k-1}; \dots; \mathbf{x}_{k-n}]$). Asociado a dicho cambio surgirán una nueva matriz de transición de estados ($\mathbf{F} = [\mathbf{F}_{k-1 \rightarrow k}, \mathbf{0}, \dots, \mathbf{0}, \mathbf{0}; \mathbf{1}, \mathbf{0}, \dots, \mathbf{0}, \mathbf{0}; \mathbf{0}, \mathbf{0}, \dots, \mathbf{1}, \mathbf{0}]$), vector de medida $\mathbf{Z} = [\mathbf{z}_k; \mathbf{z}_{k-1}; \dots; \mathbf{z}_{k-n}]$ y matriz de

medida $\mathbf{H} = [\mathbf{H}_k, \mathbf{0}, \dots, \mathbf{0}; \mathbf{0}, \mathbf{H}_{k-1}, \dots, \mathbf{0}; \mathbf{0}, \mathbf{0}, \dots, \mathbf{H}_{k-n}]$. La utilización de las expresiones del filtro de Kalman habitual sobre este nuevo sistema permite asimilar simultáneamente las medidas realizadas desde el instante $k-n$ hasta el instante k . Además, no es necesario introducir las últimas n medidas a la vez, si no que es posible introducir distintos subconjuntos de medidas, debido a las características de la matriz de ganancia de Kalman \mathbf{K} .

Sobre estas propiedades en [Challa2003] se presentan dos algoritmos para la asimilación de medidas temporalmente retardadas, en el primero se define una ventana de tiempo de tamaño n mientras que en el segundo se utiliza un estado aumentado de tamaño variable con el objeto de minimizar las necesidades de memoria del primero.

En cualquier caso, el consumo de memoria asociado a los algoritmos de [Challa2003] cuando el retardo de la medida es significativo puede ser importante, debido al tamaño de la matriz de covarianza del estado extendido. Si se supone que en el segundo algoritmo el retardo de la medida máxima es igual a n , tamaño de la ventana de tiempo del primero, los dos algoritmos de [Challa2003] son equivalentes. Suponiendo que el tamaño de la variable de estado es n_x , el algoritmo C3-2 consume $2*n*(n_x+n_x*n_x)$ unidades de memoria, mientras que los algoritmos de [Challa2003] necesitan $(n*n_x+n*n_x*n*n_x)$ unidades de memoria, por lo que nuestro algoritmo es mejor para valores elevados de n . El ahorro de memoria asociado a la segunda versión de [Challa2003] también podría ser alcanzado por nuestro algoritmo, si se sustituyese la ventana de tiempo de tamaño n , por una ventana temporal dinámica. Estas diferencias de memoria también afectarán a las necesidades reales de cómputo de cada algoritmo ya que aunque los algoritmos de [Challa2003] realizan una única etapa de asimilación de Kalman, esta es compleja debido al tamaño de la matriz de covarianza del estado extendido, mientras que nuestro algoritmo realiza después de la asimilación un conjunto de etapas de predicción sobre matrices más pequeñas.

Por último es importante destacar que en la formulación de los métodos de [Challa2003] no se considera la existencia de una señal de control en la función de transición de estados, y no se analiza la utilización de una etapa de validación de medidas. Además, se utiliza la formulación del filtro de Kalman monosensorial para la asimilación, por lo que los métodos propuestos tendrán un comportamiento subóptimo para el caso de sistemas multisensoriales.

Métodos basados en el filtro de Partículas (sistemas no lineales)

Una filosofía totalmente diferente a las anteriores es la que utiliza los filtros de Partículas, que son algoritmos capaces de representar la distribución $P(\mathbf{x}_0 : \mathbf{x}_t / \mathbf{z}_1 : \mathbf{z}_t)$ por medio de un conjunto de partículas ponderadas, para asimilar las medidas que se encuentran disponibles de forma temporalmente desordenada y retardada [Orton2001].

C3.2. *Filtros de Información para sistemas de control con sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada*

El funcionamiento de los filtros de Partículas está basado en las relaciones de independencia condicional capturadas por las redes bayesianas que modelan el problema. De los diferentes filtros de Partículas existentes, el método propuesto en [Orton2001] se fundamenta en los que trabajan con N_p partículas de la forma $\{\mathbf{x}_0^{(i)} : \mathbf{x}_t^{(i)}, w_t^{(i)}\}$, en las que se almacenan los estados desde el instante 0 hasta el instante t y su peso asociado w (que da una medida de la probabilidad relativa de la partícula). Para cada nuevo instante de muestreo $t+1$, se obtiene el nuevo estado $\mathbf{x}_{t+1}^{(i)}$ asociado a cada partícula i y se actualiza su peso asociado $w_{t+1}^{(i)}$ con la información sensorial disponible en función de las densidades de probabilidad que modelan el comportamiento de los sensores y las medidas.

En el algoritmo propuesto en [Orton2001] se supone que estado $\mathbf{x}_k^{(i)}$ asociado a la medida retardada ζ_k no será incorporado en la serie de estados de la partícula i hasta que dicha medida se encuentre disponible. Es decir, que para el instante t la partícula i se encuentra formada por los elementos $\{\mathbf{x}_0^{(i)}, \mathbf{x}_1^{(i)}, \dots, \mathbf{x}_{k-1}^{(i)}, \mathbf{x}_{k+1}^{(i)}, \dots, \mathbf{x}_t^{(i)}, w_{t/k}^{(i)}\}$. Una vez que la medida está disponible, se obtienen los N_p estados asociados a dicha medida, y se actualizan el peso asociado a cada partícula $\{\mathbf{x}_0^{(i)}, \mathbf{x}_1^{(i)}, \dots, \mathbf{x}_{k-1}^{(i)}, \mathbf{x}_k^{(i)}, \mathbf{x}_{k+1}^{(i)}, \dots, \mathbf{x}_t^{(i)}, w_t^{(i)}\}$. El cálculo de estos valores no es idéntico al cálculo de los mismos cuando la medida llega temporalmente ordenada al filtro de Partículas pero su fundamento si es similar, ya que en ambos casos se utilizan las funciones de densidad de probabilidad elegidas.

Las ventajas inherentes a los filtros de Partículas, también serán aplicables al filtro de Partículas para medidas temporalmente retardadas. Destacaremos la posibilidad de usarlo para sistemas no lineales y con diferentes tipos de relaciones probabilísticas entre las variables del sistema. A cambio, al ser un método no exacto, en el caso lineal y gaussiano se comportará como un algoritmo aproximado, en el que el orden en el que se introducen las medidas influirá de alguna forma en el resultado final obtenido. Para minimizar dicha influencia será conveniente utilizar métodos que generen partículas cuyos pesos sean similares, para evitar el predominio de unas partículas sobre otras.

En el método propuesto no se considera el problema multisensorial, la existencia de una señal de control, ni el problema de la validación. Para resolver el problema multisensorial desordenado habría que utilizar otro tipo de algoritmo de partículas y para el problema de la validación, una técnica diferente a la propuesta en los algoritmos C3-2 y C3-3, ya que para el filtro de Partículas no se estima el valor medio del estado ni su covarianza.

▪ **Comparación conjunta de los algoritmos C3-2, C3-3 y los algoritmos asociados en las subsecciones anteriores.**

Por último realizaremos una comparación conjunta entre los algoritmos C3-2, C3-3 y los algoritmos expuestos en las subsecciones anteriores, distinguiendo entre los algoritmos puramente lineales (comparación resumida en la tabla C3-10) y los que pueden ser aplicados para sistemas no lineales.

Todos los algoritmos salvo el construido sobre el filtro de Partículas, han sido desarrollados para sistemas puramente lineales, razón por la que en todo momento han sido comparados con el filtro de Información para medidas desordenadas C3-2 en vez de la versión mixta recogida en el algoritmo C3-3. La utilización de los algoritmos lineales para una versión linealizada de los sistemas es posible, pero no se obtendrán los mismos resultados que en el caso del algoritmo C3-3, cuyo comportamiento es igual al del filtro Extendido de Kalman cuando a este último se le proporcionan las medidas de forma temporalmente ordenada, salvo por las diferencias introducidas por la etapa de validación ordenada y desordenada, para los problemas en los que dicha etapa sea realizada.

Gran parte de los algoritmos lineales no recogen en la formulación del problema la existencia de una señal de control en la función de transición de estados, ni durante los ejemplos utilizados para ilustrar su comportamiento. Esta carencia no implica que el método no pueda ser utilizado para sistemas de control lineales, aunque existe la posibilidad de que sea necesario introducir alguna modificación en su formulación. El análisis de dicha posibilidad está fuera del ámbito de trabajo de esta tesis, por lo que en la tabla comparativa C3-10 distinguiremos entre los casos en los que sabemos que la señal de control ya se encuentra recogida en la formulación, y aquellos en los que no sabemos si habrá que incluir alguna modificación.

Otra suposición realizada por parte de los métodos lineales para sistemas lineales es la invertibilidad de la matriz de transición de estados \mathbf{F}_t . Como ya se planteó en el primero de los casos analizados en los que se realiza esta suposición, ésta es válida para todos los sistemas lineales discretos que se obtienen como la discretización de sistemas lineales continuos. Sin embargo, hay sistemas lineales discretos que no cumplen dicha propiedad, por lo que para estos los métodos que realizan dicha suposición no serán aplicables.

Ninguno de los métodos propuestos en la literatura utiliza una etapa de validación como la propuesta para los algoritmos C3-2 y C3-3. Aun así, la inclusión de dicha etapa en los algoritmos lineales en los que se almacenan las variables relacionadas con el estado y su covarianza (o con la información sobre el estado y su covarianza) se puede realizar fácilmente. Los resultados de la

Algoritmo	Soporte	Método	Supone F^{-1}	Formulación incluye u_t	Incluye Validación	N sensores	Óptimo	Variables almacenadas (ventana de tiempo n)
C3-2	IF	Separar Información previa y actual		√	√	√	√	$\hat{y}_{j AASIM \leq j-1}, Y_{j AASIM \leq j-1}, i_j, I_j$ y u_j con $j=t-n:t$
[Bar-Shalom2002a]	KF	Retrodicción estado	√				√	Las últimas n innovaciones o medidas
[Bar-Shalom2002b]	KF	Retrodicción estado	√					$\hat{x}_{t t-r}$ e $P_{t t-r}$ con $r=n+1:0$
[Mallick2001]	KF	Retrodicción estado	√					K_k , con $k=t-n:t$
[ZhangK2002]: AI	KF	Precálculo de variables			(1)			n subconjuntos con $\{\hat{x}_{j j}, P_{j j}, \hat{x}_{k j}, P_{k j}, T_j, W_j\}$
[ZhangK2002]: AII	KF	Precálculo de variables			(1)		√	n subconjuntos con $\{\hat{x}_{j j}, P_{j j}, M_j^*, \tilde{M}_j\}$
[Nettleton2001]	IF	Minimizar variables	√	√	(1)	√	√	$\hat{y}_{j AASIM \leq j}, Y_{j AASIM \leq j}$ y u_j con $j=t-n:t$
[ZhangK2003]: AI	KF	Minimizar variables			(1)		√	$\hat{x}_{j AASIM \leq j}, P_{j AASIM \leq j}$ con $j=t-n:t$
[ZhangK2003]: AII	KF	Minimizar variables						$P_{j AASIM \leq j}$ con $j=t-n:t$
[Challa2003]	KF	Usar variable de estados aumentada					√	X aumentada y su correspondiente P

Tabla C3-10. Comparación del filtro de Información retardada C3-2 y los otros algoritmos para sistemas lineales propuestos en la literatura. Las características que el algoritmo C3-2 comparte con el resto de los sistemas analizados han sido resaltadas en el mismo color (verde pistacho). En la nomenclatura utilizada en la tabla se distinguen entre dos algoritmos soporte (el filtro de Información (IF) y filtro de Kalman (KF)), √ significa que se cumple la propiedad propuesta en la columna, mientras que la celda permanece vacía cuando no se cumple. En el caso de la etapa de validación, las celdas marcadas con (1) significan que aunque la etapa de validación no se encuentra incluida en la formulación propuesta ésta se puede incluir fácilmente, aunque la validación no sería equivalente a la del algoritmo C3-2.

validación serán diferentes para el caso en los que se almacenen variables relacionadas con $\hat{\mathbf{y}}_{j|AASIM \leq j-1}, \mathbf{Y}_{j|AASIM \leq j-1}$ o con $\hat{\mathbf{y}}_{j|AASIM \leq j}, \mathbf{Y}_{j|AASIM \leq j}$. La diferencia se encuentra explicada en la sección de método basado en el filtro de Información y el almacenamiento mínimo de variables. En la columna de validación de la tabla comparativa C3-10, se ha marcado con (1) los métodos en los que creemos que es sencillo añadir una etapa de validación, pero que está dependerá de las variables $\hat{\mathbf{y}}_{j|AASIM \leq j}, \mathbf{Y}_{j|AASIM \leq j}$.

Los métodos basados en el filtro de Información mantienen su condición de óptimo para el caso multisensorial, ya que en dicha formulación la fusión multisensorial se realiza de la misma forma que la fusión monosensorial. Sin embargo, los métodos basados en el filtro de Kalman recogidos en la literatura no contemplan el caso multisensorial. Debido a las diferencias existentes entre las expresiones de asimilación monosensorial y multisensorial (que se encuentran recogidas en el apéndice I), cuando se aplican directamente los métodos óptimos basados en el filtro de Kalman al caso multisensorial éstos perderán su condición de óptimo para convertirse en métodos subóptimos.

En la última columna de la tabla C3-10 se muestran las variables que hay que almacenar para cada uno de los métodos para sistemas lineales cuando se supone que puede llegar una medida retardada en cualquier instante de muestreo dentro de una ventana de tiempo de tamaño \mathbf{n} .

La comparación de la complejidad computacional de cada uno de los métodos no ha sido incluida, ya que las diferencias existentes entre los mismo hace necesaria su implementación para poder realizar un estudio en profundidad de la misma. Sin embargo, para el método más parecido al nuestro, el propuesto en [Nettleton2001], en la subsección donde dicho método es presentado se concluyó que nuestro método tenía una complejidad menor mientras que el de [Nettleton2001] necesitaba menos memoria.

La comparación del único método no lineal con el filtro Mixto de Información C3-3 no se recoge en ninguna tabla, debido a las diferencias fundamentales existentes entre ambos métodos. Nuestro método se fundamenta en el filtro de Información Extendido multisensorial, que permite estimar el estado sistemas de control multisensoriales no lineales con errores gaussianos, mientras que el método de [Challa2003] se fundamenta en un filtro de Partículas para sistemas monosensoriales cuyas variables se relacionan por expresiones de probabilidad de cualquier tipo.

Por último es importante destacar el origen bayesiano común de la filosofía utilizada para los métodos propuestos en [Challa2003], [Orton2001] y en los algoritmos C3-2 y C3-3. En los cuatro casos se utiliza como punto de partida un análisis puramente bayesiano del problema, que se puede fundamentar sobre la estructura de la red bayesiana dinámica como la propuesta al inicio este capítulo. A pesar del origen común del análisis que los fundamenta, los algoritmos finalmente desarrollados se han construido sobre algoritmos diferentes.

Capítulo IV

Análisis del funcionamiento de los algoritmos de filtrado propuestos

En este capítulo se realiza un análisis exhaustivo del funcionamiento de los principales algoritmos presentados en los capítulos anteriores. Los algoritmos elegidos para este análisis son: la versión del algoritmo de inferencia exacta basado en el árbol de unión para la red bayesiana dinámica del capítulo III, que minimiza el número de operaciones e incluye la etapa opcional de validación; el algoritmo de Información Asíncrono para sistemas puramente lineales capaz de asimilar correctamente medidas retardadas y desordenadas; y el algoritmo Extendido de Información Asíncrono para sistemas mixtos (con funciones lineales y/o no lineales) capaz de asimilar correctamente medidas retardadas y desordenadas. Es decir, los algoritmos del capítulo anterior identificados bajo los epígrafes C3-1, C3-2 y C3-3. Estos tres algoritmos, que identificaremos respectivamente como Bayes, IFAsyn e IFAsynMixto, también se encuentran recogidos, a modo de resumen, en el apéndice II.

Además, para destacar la importancia de la asimilación de la información retardada, se compararan los resultados obtenidos por estos tres algoritmos con los extraídos por las versiones habituales del algoritmo de Kalman multisensorial y su versión extendida, cuyas expresiones se encuentran recogidas en las tablas A1-8 y A1-10 del apéndice I.

El estudio, realizado sobre varios sistemas, comprende un análisis de los resultados obtenidos por los distintos algoritmos en el que se pone de manifiesto diferentes características de los algoritmos C3-1, C3-2 y C3-3, y se examina el consumo de memoria y las necesidades de cómputo de los mismos.

El análisis realizado se encuentra dividido en dos partes diferenciadas, de acuerdo con el tipo de sistemas sobre los que se puede aplicar cada algoritmo: en la primera se recoge un estudio sobre el comportamiento de los algoritmos aplicables a los sistemas lineales, mientras que en la segunda se comprueba el funcionamiento de los algoritmos aplicables a sistemas con alguna ecuación de modelo (de transición y/o medida) no lineales. Además, se utilizarán las ecuaciones de cada sistema para: (1) generar, mediante simulación, las medidas proporcionadas por cada sensor, y (2) estimar, dentro del algoritmo seleccionado, los valores del estado y de la covarianza para cada instante de muestreo.

En la primera parte de este capítulo se analizan los resultados obtenidos por los algoritmos de Bayes, IFAsyn y de Kalman habitual (tabla A1-8) sobre un sistema lineal sencillo. En dichos resultados se observan las diferencias y similitudes de los tres algoritmos, la influencia del desorden de las medidas en los resultados obtenidos, la importancia de la existencia de la etapa de validación cuando hay medidas erróneas, y la evolución del valor del estado y covarianza para cada instante de muestreo según se va asimilando la información medida en instantes anteriores y que llega al algoritmo de forma temporalmente retardada. Además se examina el consumo de memoria y las necesidades de cálculo de los tres algoritmos ante diferentes condiciones experimentales,

como son el retardo de las medidas y la decisión de ejecutar los algoritmos con o sin etapa de validación.

En la segunda parte se estudian los resultados obtenidos por los algoritmos IFAsynMixto y Kalman Extendido (tabla A1-10) sobre varios sistemas diferentes. Para examinar el consumo de memoria y las necesidades de cómputo se utilizará el sistema lineal elegido para el análisis de los algoritmos lineales, sintonizando el algoritmo IFAsynMixto de forma que trate a los sensores lineales como si no lo fuesen, para evitar que la sobrecarga computacional debida a diferentes ecuaciones no lineales sea contabilizada en el análisis comparativo. Para estudiar los resultados obtenidos por los dos algoritmos se utilizan dos sistemas no lineales diferentes. Durante el análisis de los resultados obtenidos para los dos sistemas se pondrá de manifiesto la influencia del desorden de las medidas en los resultados obtenidos y en la evolución de los mismos según se asimilan las medidas desordenadas, la necesidad de término corrector de la medida para la asimilación sensorial proveniente de sistemas no lineales y la necesidad del recálculo de la información sensorial asociada a los sensores no lineales cuando llega al algoritmo información retardada proveniente de cualquier sensor. Durante este último estudio se utilizará, además del algoritmo IFAsynMixto, una versión modificada del mismo, recogida en la tabla A2-2 del apéndice II, denominada IFAsynMixtoEsp, y que se caracteriza por no recalcular la información sensorial asociada a los sensores no lineales. Comparando los resultados obtenidos por los dos algoritmos (IFAsynMixto e IFAsynMixtoEsp) sobre los dos sistemas no lineales propuestos se observará que en algunos casos el recálculo de la información sensorial asociada a los sensores no lineales es imprescindible, mientras que en otros puede considerarse innecesaria.

Para llevar a cabo el análisis exhaustivo recogido en este capítulo, tanto los algoritmos propuestos (Bayes, IFAsyn, IFAsynMixto e IFAsynMixtoEsp) como el algoritmo de Kalman clásico (en su versión normal y extendida) han sido implementados en Matlab (versión 5.3 y posteriores) bajo la forma de una herramienta (librería) fácilmente configurable. Dicha herramienta permite estimar el estado de cualquier sistema modelable con funciones de transición de estado y de medida que incluyen términos aditivos de ruido gaussiano, tras la definición del conjunto de funciones de Matlab que implementan las ecuaciones ideales del sistema y de medida, y de las matrices de covarianza asociadas al ruido.

Como ya se ha mencionado a lo largo de esta introducción este capítulo se encuentra organizado en dos secciones. En la primera se analiza el funcionamiento de los algoritmos para sistemas con modelos lineales mientras en la segunda se analiza el funcionamiento de los algoritmos para sistemas con modelos no lineales.

C4.1. Análisis del funcionamiento de los algoritmos para sistemas con modelos lineales

En esta sección se estudian distintos aspectos del funcionamiento de los algoritmos presentados en los capítulos anteriores que funcionan únicamente sobre sistemas lineales. De todos los algoritmos propuestos, sólo se analizará el funcionamiento de los más generales. Esto es, para el algoritmo exacto de inferencia de Bayes se ha elegido el algoritmo C3-1 (ya que es la versión desarrollada para la red dinámica que permite incluir una etapa de validación y la que minimiza el consumo de memoria y el número de operaciones que hay que realizar) mientras que para el algoritmo de Información se ha seleccionado el algoritmo C3-2 (porque es la versión propuesta para el caso puramente lineal). Estos dos algoritmos, que identificaremos como Bayes e IFAsyn respectivamente, también se encuentran recogidos, a modo de resumen, en el apéndice II.

Durante el análisis realizado, además de estos dos algoritmos, también se utilizará el algoritmo de Kalman multisensorial habitual (cuyas etapas se recogen en la tabla A1-8 del apéndice I), que desechará las medidas que le lleguen de forma temporalmente retardadas. Este algoritmo será identificado como Kalman a lo largo de este capítulo.

Para realizar el estudio de este apartado, se utilizará un sistema lineal dinámico muy sencillo, cuyo vector de estados está formado por tres componentes, subconjuntos de las cuales son medidas, de forma directa, por tres sensores diferentes. Las expresiones del modelo del sistema y sensorial elegido servirán para: (1) generar, mediante simulación y a partir de un valor del estado inicial y los valores de la señal de control, las medidas de los sensores; y (2) obtener, a partir de dicho estado inicial, de la señal de control y de las medidas de los sensores, el valor estimado del estado por los diferentes algoritmos.

Se examinará el comportamiento de los tres algoritmos (Bayes, IFAsyn y Kalman) bajo diferentes condiciones experimentales, como son el funcionamiento ante diferentes retardos en la recepción de las medidas y la existencia de medidas erróneas cuando se incluye o elimina la etapa de validación. Este análisis comprenderá estudios comparativos de los resultados obtenidos por los diferentes algoritmos y de sus necesidades de memoria y cálculo.

El estudio se encuentra dividido en tres secciones. En la primera se presenta el modelo del sistema y de los sensores elegidos y se describen los datos experimentales sobre los que se fundamentan todos los resultados de este apartado. En la segunda se realiza una serie de estudios comparativos de los resultados obtenidos por cada algoritmo con el objeto de resaltar las diferencias y similitudes existentes entre los mismos. Y por último, se analizan el consumo de memoria y las necesidades de cálculo de los algoritmos para diferentes ejemplos.

C4.1.1. Descripción general del sistema y de las pruebas realizadas

Para ilustrar el funcionamiento de los algoritmos se ha elegido un sistema muy sencillo, cuyo vector de estado está formado por tres componentes. Grupos de estas componentes, que pueden ser interpretados como la posición, x e y , y la orientación, θ , de un robot, son observadas de forma directa por tres sensores lineales.

A partir de las expresiones del modelo de transición y sensorial del sistema (que será presentado y analizado a continuación), de un estado inicial y del valor de la señal de control durante un intervalo de tiempo, se generarán, mediante simulación, los datos experimentales básicos para todos los experimentos. Estos datos están compuestos por una única trayectoria (x , y , θ) y las medidas realizadas para cada instante de muestreo por cada sensor sobre dicha trayectoria. La trayectoria generada durante el proceso de simulación será almacenada para poder compararla con la trayectoria estimada por los algoritmos a partir de la posición inicial, las medidas de los sensores, y la señal de control aplicada durante su generación.

Por último, los algoritmos serán sometidos a diferentes pruebas, generadas todas a partir de la misma información básica. La diferencia radicará en los retardos de recepción de las medidas, en la modificación de una medida durante un intervalo de tiempo para simular el funcionamiento erróneo de un sensor y comprobar la importancia de la validación en estos casos, en la falta de medidas de un sensor durante un intervalo de tiempo, etc.

En las tres secciones siguientes se presentarán el modelo elegido, la información básica generada para todos los experimentos, y los conjuntos de pruebas utilizados para comprobar el correcto funcionamiento de los algoritmos.

▪ Modelo del sistema y de los sensores

Para ilustrar el funcionamiento de los algoritmos se ha elegido un sistema cuyo vector de estado está formado por tres elementos, que pueden ser interpretados como la posición, x e y , y la orientación, θ , de un robot, y cuyos sensores lineales, tres en total, observan directamente subconjuntos de dichos estados. Las expresiones del modelo ideal (sin términos aditivos de ruido) se recogen en la primera columna de la tabla C4-1, mientras que las ecuaciones del modelo real, expresadas en forma vectorial, y los términos relacionados con las expresiones vectoriales se recogen en las dos columnas siguientes.

En el modelo de transición propuesto para el sistema, cuyas ecuaciones se encuentran recogidas en la parte superior de la tabla C4-1, el valor de cada una de las componentes del vector estado en el instante $t+1$ es igual a la suma del valor de dicho estado en el instante anterior t , del valor de una señal de control, y del valor de una variable aleatoria gaussiana con media cero y covarianza fija.

Además, como la matriz de covarianza \mathbf{Q}_t elegida es de la forma $\mathbf{Q}_t(i, j) = \mathbf{Q}_{i,j} \delta_{i,j}$, y en la ecuación de transición de estados no aparecen relacionados unos estados con otros, los tres estados del sistema son linealmente independientes, propiedad que nos permitirá analizar el comportamiento de los algoritmos de fusión presentados en el capítulo III con mayor facilidad.

Las ecuaciones de medida se muestran en las tres últimas filas de la tabla C4-1. El primer sensor, por ejemplo una brújula, observa únicamente la tercera componente del vector de estado (θ). El segundo, que podría ser un sistema triular de ultrasonidos, mide directamente los valores de cada una de las componentes del vector (x, y, θ). Y el tercero, por ejemplo un GPS, recoge información sobre la posición del sistema (x, y) y no sobre su orientación. En las expresiones reales aparecen términos aditivos de ruido, cuya covarianza se encuentra representada en las matrices $\mathbf{R}_{s,t}$ de la tercera columna. Es importante destacar que ninguno de los sensores devuelve medidas formadas como la combinación lineal de dos o más estados. Además el sensor2 mide directamente los 3 estados mientras que el sensor1 y el sensor3 miden dos subconjuntos disjuntos de los mismos. Las características de los sensores elegidos también facilitará el análisis posterior del comportamiento de los algoritmos de fusión.

	Modelo Ideal	Modelo Real	con $\mathbf{x}_t = (x(t) \ y(t) \ \theta(t))^T$ y
Sistema	$x(t+1) = x(t) + u_x(t)$ $y(t+1) = y(t) + u_y(t)$ $\theta(t+1) = \theta(t) + u_\theta(t)$	$\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{u}_t + \mathbf{v}_t$	$\mathbf{F}_t = \mathbf{I}_{3 \times 3}$, $\mathbf{u}_t = \begin{pmatrix} u_x(t) \\ u_y(t) \\ u_\theta(t) \end{pmatrix}$, $\mathbf{Q}_t = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.2\pi/180 \end{pmatrix}^2$
Sensor1	$z_{1\theta}(t+1) = \theta(t+1)$	$\mathbf{z}_{1,t+1} = \mathbf{H}_{1,t+1} \mathbf{x}_{t+1} + \mathbf{v}_{1,t+1}$	$\mathbf{H}_{1,t+1} = (0 \ 0 \ 1)$, $\mathbf{z}_{1,t+1} = z_{1\theta}(t+1)$, $\mathbf{R}_{1,t+1} = (\pi/180)^2$
Sensor2	$z_{2x}(t+1) = x(t+1)$ $z_{2y}(t+1) = y(t+1)$ $z_{2\theta}(t+1) = \theta(t+1)$	$\mathbf{z}_{2,t+1} = \mathbf{H}_{2,t+1} \mathbf{x}_{t+1} + \mathbf{v}_{2,t+1}$	$\mathbf{H}_{2,t+1} = \mathbf{I}_{3 \times 3}$, $\mathbf{z}_{2,t+1} = \begin{pmatrix} z_{2x}(t+1) \\ z_{2y}(t+1) \\ z_{2\theta}(t+1) \end{pmatrix}$, $\mathbf{R}_{2,t+1} = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 2\pi/180 \end{pmatrix}^2$
Sensor3	$z_{3x}(t+1) = x(t+1)$ $z_{3y}(t+1) = y(t+1)$	$\mathbf{z}_{3,t+1} = \mathbf{H}_{3,t+1} \mathbf{x}_{t+1} + \mathbf{v}_{3,t+1}$	$\mathbf{H}_{3,t+1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $\mathbf{z}_{3,t+1} = \begin{pmatrix} z_{3x}(t+1) \\ z_{3y}(t+1) \end{pmatrix}$, $\mathbf{R}_{3,t+1} = \begin{pmatrix} 0.05 & 0 \\ 0 & 0.05 \end{pmatrix}^2$

Tabla C4-1. Ecuaciones del modelo del sistema y de los sensores para el caso lineal. En la primera columna, aparecen las ecuaciones con sus componentes de forma explícita, para el modelo ideal (sin ruido). En la segunda columna se muestran las ecuaciones vectoriales reales, a las que se le ha añadido el término aditivo de ruido blanco. En la tercera columna se representan las matrices y vectores que relacionan el modelo de la primera y segunda columna (salvo por el término aditivo de ruido) y la covarianza asociada al ruido para cada caso (\mathbf{Q}_t y $\mathbf{R}_{s,t}$).

▪ Datos experimentales básicos

Los datos experimentales utilizados durante los distintos ejemplos han sido generados, mediante simulación, a partir de las ecuaciones de transición y de medida de la tabla C4-1, de un valor inicial para estado, y de los valores de la señal de control, aplicada al móvil, cada décima de segundo,

durante un intervalo de tiempo de 60 segundos. De esta forma, y con el objeto de facilitar el análisis de los resultados del algoritmo para distintos retardos y comportamiento de los sensores, se generará una única trayectoria (x, y, θ) y las medidas sensoriales básicas asociadas a la misma.

La trayectoria generada para los experimentos, almacenada únicamente para poder compararla posteriormente con las trayectorias estimadas por los diferentes algoritmos en las diferentes pruebas, se representa en las gráficas de la figura C4-1. La gráfica superior izquierda representa el estado y frente al estado x , tanto para el caso ideal (sin incluir el término aditivo aleatorio de ruido v_t) como para el caso real (añadiendo dicho término). La gráfica superior derecha representa la orientación θ frente al tiempo, también para ambos casos. La gráfica inferior izquierda recoge una representación de los tres estados frente al tiempo y la gráfica inferior derecha representa la diferencia entre el caso real y el ideal para cada uno de los estados. Es importante destacar que en la gráfica (c), y en otras posteriores en las que se representen diferentes valores del mismo estado frente al tiempo, las diferencias en orientaciones suelen ser directamente apreciables (parte inferior de la gráfica (c)), mientras que la diferencias en cualquiera de los otros estados pueden no serlo (parte superior e intermedia de la gráfica (c)). En los casos en lo que esto ocurra, la representación de diferencias de valores de estados frente al tiempo (gráfica (d)) resultará ventajosa.

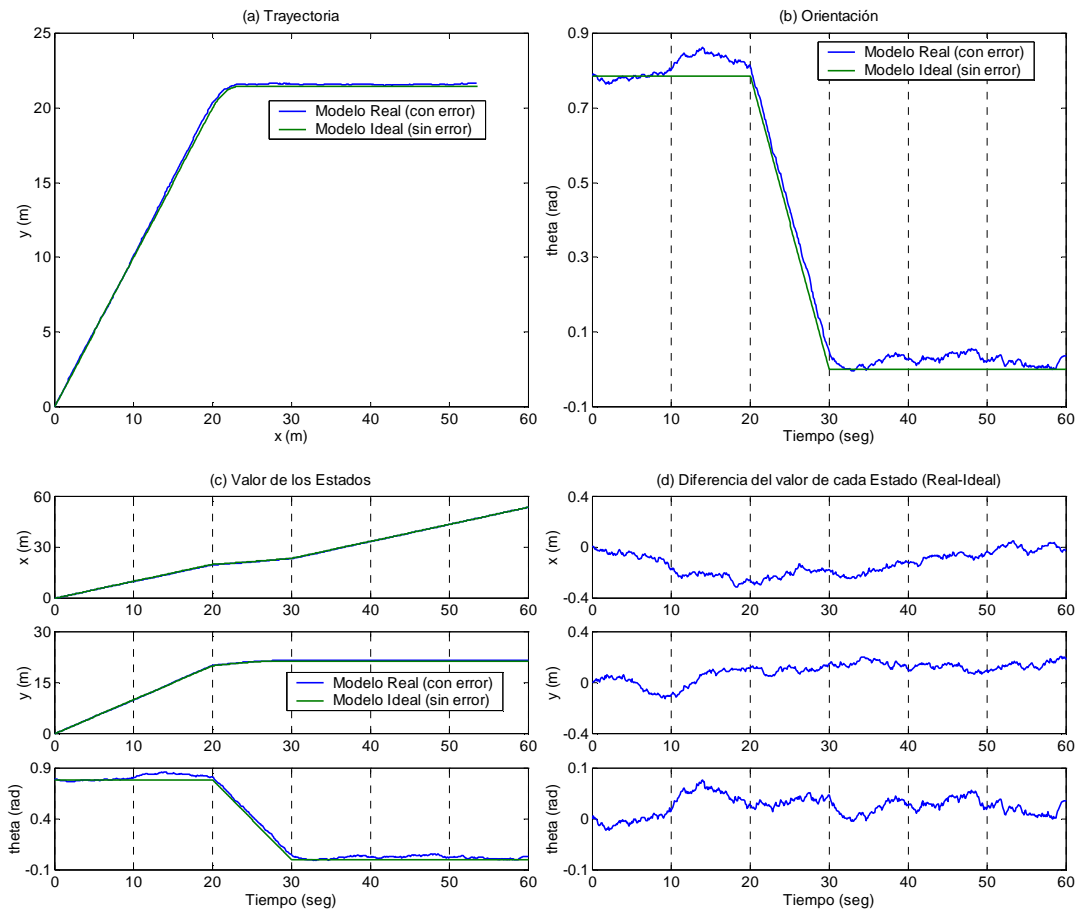


Figura C4-1. Trayectoria y Orientación utilizada durante todos los experimentos de los sistemas lineales.

A partir de la trayectoria real, y haciendo uso de las expresiones de modelo de medida reales, se han generado los valores de las medidas asociadas a los distintos instantes de muestro, las cuales son utilizadas, tal y como se generaron o tras alguna modificación, en los ejemplos prueba propuestos en la sección siguiente.

▪ **Descripción de las Pruebas Experimentales**

Para comprobar el funcionamiento de los algoritmos Bayes (C3-1), IFAsyn (C3-2) y Kalman (A1-8) se han realizado dos tipos de pruebas experimentales: uno orientado a comprobar la influencia en el valor estimado del retardo de la medida y del funcionamiento de los sensores, que pueden realizar medidas erróneas o no medir durante un determinado intervalo de tiempo, y otro destinado a analizar el consumo de memoria y las necesidades de cálculo en función del retardo máximo existente para cada uno de los algoritmos.

Grupo Experimental I: Influencia en los algoritmos de la existencia de retardos y errores en el funcionamiento y medidas de los sensores

Para analizar la respuesta de los algoritmos de Bayes e IFAsyn bajo distintas condiciones experimentales se utiliza un banco de pruebas formada por seis experimentos diferentes, cuyas características se esquematizan en las gráficas de la figura C4-2. En cada una ellas se representa, para los tres sensores, el retardo existente en la recepción de la medida (eje de ordenadas) frente al instante en el que la medida es capturada (eje de abscisas). El instante de recepción de la medida se puede calcular sumándole al instante en el que la medida es capturada el retardo que existe en la recepción.

Los seis experimentos propuestos se pueden dividir en tres subconjuntos (uno por cada fila de la figura) cuyos experimentos únicamente se diferencian por la falta (primera columna de la figura) o existencia (segunda columna de la figura) de retardo en la recepción de las medidas. Los retardos introducidos en los tres experimentos en los que éste fenómeno está presente (segunda columna de la figura) son iguales, y consisten en que las medidas del sensor1 capturadas entre los 20 y 40 segundos llegan 1 segundo retardadas, mientras que el mismo retardo (1 segundo) es introducido en las medidas capturadas por el sensor2 entre los 30 y 50 segundos, y por el sensor3 entre los 10 y 50 segundos.

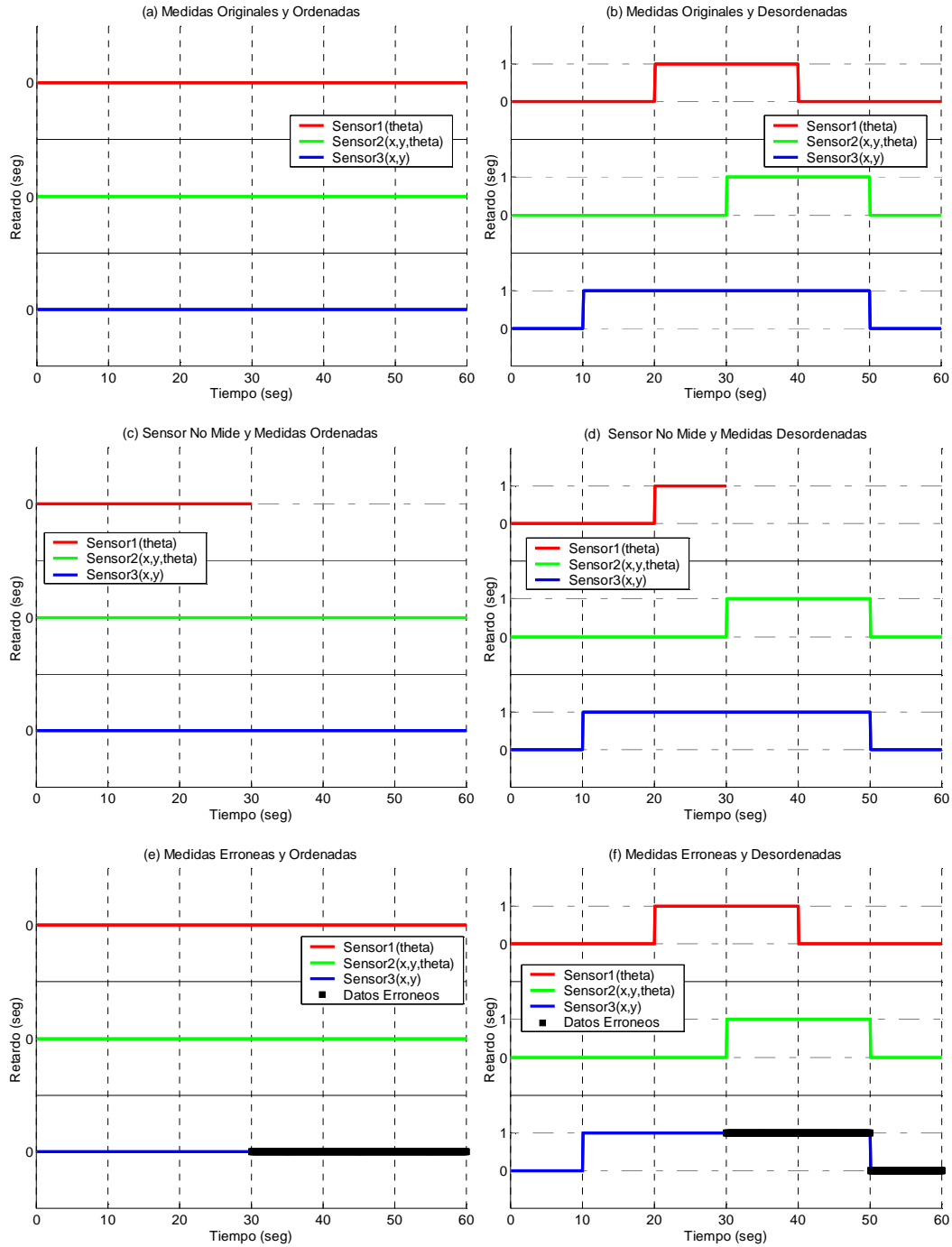


Figura C4-2. Caracterización de los experimentos del Grupo Experimental I: Influencia de la existencia de retardos y errores en el funcionamiento y medidas de los sensores.

Los experimentos de cada uno de los tres subconjuntos (representados en cada una de las filas de la C4-2) se diferencian en el funcionamiento de alguno de sus sensores:

- Medidas originales (fila superior): se utilizan todas las medidas originales generadas durante la simulación. De esta forma se puede comprobar el funcionamiento de los algoritmos sin

que falten medidas ni que éstas contengan más información errónea que la introducida por los errores de medida modelados.

- Sensor que no mide (fila intermedia): se utilizan todas las medidas originales generadas durante la simulación salvo aquellas proporcionadas por el sensor1 a partir de los 30 segundos. De esta forma se observará el funcionamiento del algoritmo cuando uno de los sensores deje de medir.
- Medidas erróneas (fila inferior): se utilizan todas las medidas originales generadas durante la simulación salvo las proporcionadas por el sensor3 a partir de los 30 segundos, que son modificadas mediante la suma de un valor lo suficientemente grande para que dichas medidas sean consideradas erróneas por la etapa de validación de los algoritmos.

Sobre los 6 experimentos propuestos se ejecutarán dos veces los tres algoritmos (Bayes, IFAsyn y Kalman), una vez incluyendo la etapa de validación y otra sin dicha etapa. Por lo tanto, para cada algoritmo se realizarán 12 experimentos diferentes (6x2 posibilidades de validación), lo que nos permitirá analizar la influencia que tienen sobre los mismos los retardos, el funcionamiento de los sensores y la utilización de la etapa de validación.

El análisis realizado sobre este grupo experimental comprenderá análisis comparativos de los valores estimados por los algoritmos, y del consumo de memoria y necesidades de cómputo para cada uno de los 12 casos.

Grupo Experimental II: Influencia de la duración del retardo en el consumo de memoria y número de operaciones

En el grupo experimental I el retardo introducido en la recepción de las medidas únicamente toma dos valores, 0 ó 1, por lo que dichos experimentos no nos permiten realizar un análisis en profundidad de las necesidades de cómputo y de consumo de memoria para diferentes retardos. Para conseguirlo, se utilizará un grupo experimental adicional, cuyas características se esquematizan en las gráficas de la figura C4-3, en el que únicamente se introducen retardos diferentes sobre las medidas originales.

Los experimentos del grupo, un total de 18, se pueden dividir en dos subconjuntos de 9 experimentos. En el primero, representado en la gráfica de la izquierda, se introduce únicamente el retardo variable **R** en el sensor3, mientras que en el segundo, representado en la gráfica de la derecha, el retardo variable **R** es introducido en todos los sensores. En ambos casos, el retardo variable **R** tomará valores en el rango de 0 a 8 segundos, con una diferencia de 1 segundo entre dos valores sucesivos.

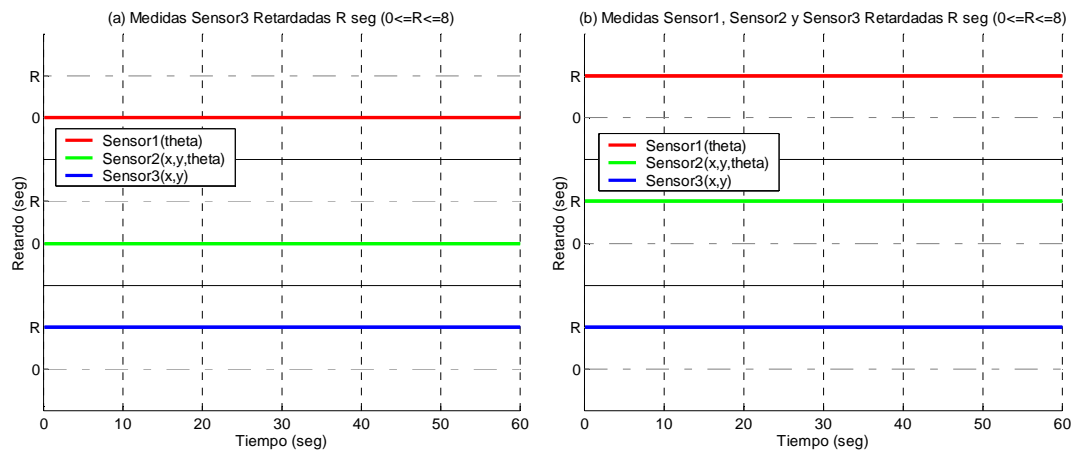


Figura C4-3. Caracterización de los experimentos del Grupo Experimental II: Influencia de la duración del retardo en el consumo de memoria y número de operaciones

El análisis realizado sobre este grupo experimental se centrará solamente en la comparación del consumo de memoria y de las operaciones realizadas por los tres algoritmos para los diferentes casos planteados.

C4.1.2. Análisis de los resultados

En este apartado se presenta el análisis comparativo de los resultados obtenidos por los algoritmos Bayes (C3-1), IFAsyn (C3-2) y Kalman (A1-8) sobre los grupos experimentales presentes en la sección anterior. Dicho estudio ha sido dividido en dos secciones: en la primera se analizan las diferencias existentes en los valores estimados por los diferentes algoritmos, y en la segunda se lleva a cabo un examen de las necesidades de cómputo y de memoria de los algoritmos. El primer tipo de análisis es realizado únicamente sobre el grupo experimental I, mientras que el segundo se lleva a cabo sobre los experimentos propuestos en los grupos experimentales I y II.

▪ Análisis comparativo de los valores estimados por los diferentes algoritmos (grupo experimental I)

Los 12 (6×2 posibilidades de validación) experimentos propuestos en el grupo experimental I permiten realizar diferentes estudios comparativos sobre los resultados (valores estimados del estado y sus covarianzas) obtenidos por los algoritmos de Bayes, IFAsyn y Kalman, y comprobar mediante dichos análisis la influencia de la existencia de retardos, la necesidad de la validación para rechazar medidas cuando los sensores proporcionan medidas erróneas, la equivalencia existente entre los algoritmos de Bayes e IFAsyn, etc.

El análisis comparativo de este apartado se encuentra dividido en cuatro secciones. En el primero se analiza, para cada uno de los 12 experimentos propuestos, los resultados obtenidos por los tres algoritmos. A continuación se observa la influencia del desorden de las medidas en los resultados,

destacando en este análisis, la influencia del desorden de las medidas en la etapa de validación. Posteriormente se justifica la necesidad de la etapa de validación, en los casos en los que los sensores proporcionan medidas cuyo error no se encuentra modelado en las expresiones sensoriales. Y por último se analiza como evoluciona el valor estimado para un instante de muestreo, según se van asimilando las medidas que llegan temporalmente retardadas.

Comparación de los resultados obtenidos para el mismo ejemplo por los tres algoritmos

Con el objeto de comprobar la equivalencia existente entre el algoritmo de Bayes e IFAsyn, y resaltar su capacidad para trabajar con medidas temporalmente desordenadas, se compararán los resultados (valor estimado del estado y su covarianza) obtenidos por los tres algoritmos (Bayes, IFAsyn y Kalman) en todos los instantes de muestreo. Además, también se comprobará si el mismo número de medidas ha sido incorporado como evidencia en los algoritmos o si ha sido rechazado, bien porque las medidas han llegado tarde y no han podido ser procesadas (en el caso de Kalman), bien porque han sufrido una suerte diferente durante el proceso de validación.

Para realizar esta comparación, obtendremos la diferencia máxima existente entre los valores estimados del estado (y de la varianza) para cada una de las tres posibles combinaciones diferentes de parejas de algoritmos. Para el análisis relativo a la suerte sufrida por las medidas, se contabilizará el número de las mismas que han sido tratadas de forma diferente. En la siguiente tabla se recoge, para cada uno de los experimentos, los resultados de esta comparativa.

Experimento		IFAsyn vs. Bayes			IFAsyn vs. Kalman			Kalman vs. Bayes		
		Estado	Var.	Valid.	Estado	Var.	Valid.	Estado	Var.	Valid.
Sin retardo	aSv	3.6e-14	3.8e-19	0	5.6e-16	1.4e-20	0	3.6e-14	3.8e-19	0
	aCv	7.8e-14	4.9e-19	0	6.6e-16	2.7e-20	0	7.8e-14	4.9e-19	0
	cSv	3.6e-14	6.6e-19	0	5.6e-16	1.4e-20	0	3.6e-14	6.6e-19	0
	cCv	8.5e-14	8.1e-19	0	6.7e-16	2.7e-20	0	8.5e-14	8.1e-19	0
	eSv	4.3e-14	3.8e-19	0	5.6e-16	1.4e-20	0	4.3e-14	3.8e-19	0
	eCv	2.0e-13	3.5e-18	0	6.7e-16	2.7e-20	0	2.0e-13	3.5e-18	0
Con retardo	bSv	3.6e-14	3.8e-19	0	1.4e-1	2.1e-2	800	1.4e-1	2.1e-2	800
	bCv	7.8e-14	4.5e-19	0	1.6e-1	2.1e-2	802	1.6e-1	2.1e-2	802
	dSv	3.5e-14	6.6e-19	0	1.4e-1	2.1e-2	700	1.4e-1	2.1e-2	700
	dCv	8.5e-14	8.1e-19	0	1.6e-1	2.1e-2	702	1.6e-1	2.1e-2	702
	fSv	4.3e-14	3.8e-19	0	8.1	2.1e-2	800	8.1	2.1e-2	800
	fCv	2.0e-13	3.7e-18	0	1.5e-1	2.0e-2	803	1.5e-1	2.0e-2	803

Tabla C4-2. Tabla comparativa de los resultados obtenidos por los diferentes algoritmos para cada uno de los 12 ejemplos. En la primera columna se representan los experimentos, agrupados en dos subconjuntos (sin y con retardo) e identificados con tres letras: la primera indica el experimento de la figura C4-2 a la que se refieren los resultados (*a* medidas originales ordenadas, *b* medidas originales desordenadas, *c* sensor no mide y medidas ordenadas, *d* sensor no mide y medidas desordenadas, *e* medidas erróneas y ordenadas, *f* medidas erróneas y desordenadas), y las otras dos sirven para diferenciar si el algoritmo se ha ejecutado sin etapa de validación (Sv) o con ella (Cv). Para cada par de algoritmos (A vs. B) se representa la diferencia máxima existente entre los valores del estado estimado por cada uno, la diferencia máxima existente entre los valores de la varianza, y el número de medidas que han sido tratadas de forma diferente por cada algoritmo. En verde se resalta, para cada experimento, la pareja de algoritmos que obtienen los resultados más parecidos, mientras que en naranja se resaltan los casos en los que la diferencia no puede considerarse despreciable.

En la comparación del filtro de Información asíncrono para medidas retardadas (IFAsyn) frente al algoritmo exacto de inferencia optimizado (Bayes), recogida en la columna IFAsyn vs. Bayes, se puede observar que ambos filtros son equivalentes ya que utilizan la misma información sensorial (*Valid* es 0 para todos los ejemplos) y la diferencia máxima de los estados y de la varianza a lo largo de toda la trayectoria es despreciable. El origen de esta diferencia insignificante se encuentra en los diferentes errores de cálculo (redondeo) introducidos por los algoritmos al ser ejecutados.

En la comparación del filtro de Kalman habitual con los otros dos algoritmos (IFAsyn vs. Kalman, Kalman vs. Bayes), se puede observar que los resultados obtenidos por los tres algoritmos son equivalentes para los ejemplos en los que no existe retardo en la comunicación. Además, en la tabla se observa que para los casos sin retardos la semejanza es mayor entre los resultados obtenidos por el filtro de Kalman e IFAsyn, que entre Bayes y cualquiera de los otros dos algoritmos.

Sin embargo y como cabía esperar, la equivalencia existente entre los tres filtros para el caso en el que no existe retardo en la comunicación desaparece en los ejemplos en los que existen medidas retardadas. Esto se debe a que el filtro de Kalman no procesa la información sensorial asociada a las medidas retardadas, que son directamente desechadas. El número de medidas que son tratadas de forma diferente son aquellas que llegan retardadas (800 para los ejemplos *b* y *f*, y 700 para el *d*) más las pocas medidas que han sufrido una suerte diferente durante el proceso de validación.

Para el ejemplo *fSv* la diferencia máxima en los estados entre Kalman y los otros dos algoritmos es mucho mayor que para el resto de los casos, ya que es el ejemplo de medidas erróneas y retardadas sin etapa de validación, caracterizado en la gráfica (f) de la figura C4-2. Durante el mismo, hay medidas erróneas con y sin retardo, y como los algoritmos no utilizan la etapa de validación y no detectan el fallo del sensor procesarán todas las que puedan: Bayes e IFAsyn procesarán todas las medidas erróneas (es decir, las medidas del sensor3 desde los 30 segundos), mientras que el filtro de Kalman únicamente procesará aquellas que no lleguen retardas (es decir, las medidas erróneas del sensor3 desde los 50 segundos). Como consecuencia de esta diferencia, el sensor3 confunde las posiciones (*x* e *y*) obtenidas por Bayes e IFAsyn a partir de los 30 segundos, mientras que esa confusión no es introducida en los valores estimados por el filtro de Kalman hasta después de los 50 segundos. En las gráficas de la figura C4-4 se puede observar este comportamiento. En ella se representan, para este caso particular, (a) el valor real de cada estado y los valores estimados por los tres algoritmos para cada estado frente al tiempo (IFAsyn y Bayes coinciden), y (b) la diferencia entre los valores estimados por cada algoritmo y el real (IFAsyn y Bayes coinciden). Aunque podría parecer que para este ejemplo particular el comportamiento del filtro de Kalman es mejor que el de los otros dos, no debe extraerse ninguna conclusión de esta observación, ya que si las medidas erróneas no hubiesen llegado tarde, mientras que medidas correctas de los otros sensores sí, el funcionamiento de Kalman sería peor que el de los algoritmos Bayes e IFAsyn.

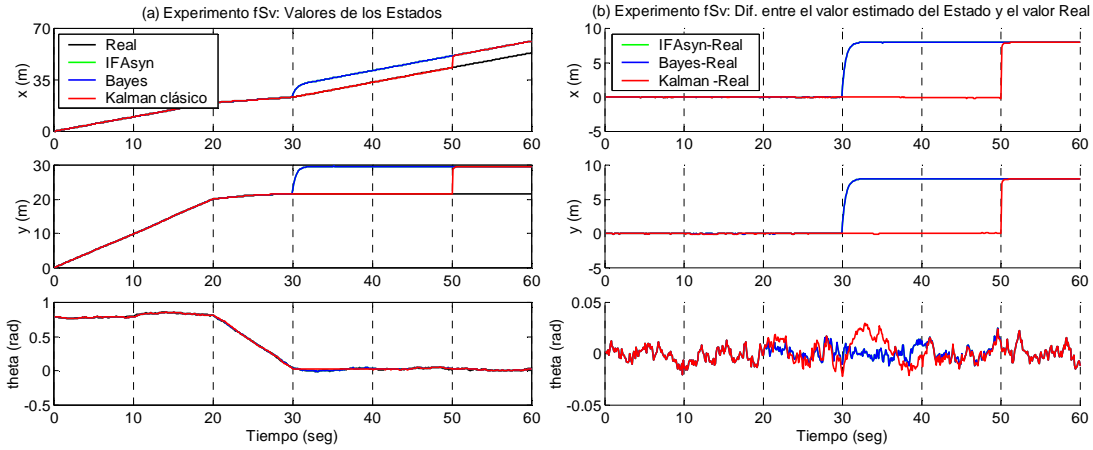


Figura C4-4. Valores estimados del estado por cada uno de los filtros, y diferencia entre los valores estimados y el real, para el caso con medidas erróneas y retardadas, cuando no se usa etapa de validación.

En el resto de los ejemplos con medidas desordenadas, la diferencia entre los estados estimados por el filtro de Kalman y los otros dos filtros es mucho menor, a pesar de existir, ya que para las medidas que no utiliza Kalman porque son recibidas temporalmente retardadas por el algoritmo, el filtro utiliza los valores de la predicción. Para el sistema elegido, estos valores no se ven modificados de forma sustancial con la introducción de la información de medidas correctas (para el caso fCv, las medidas erróneas serán desechadas en todos los filtros debido a la etapa de validación). En el caso de la varianza, la diferencia máxima entre la calculada con Kalman y los otros dos algoritmos es mucho mayor, ya que la falta de las medidas rechazadas por Kalman por haber llegado tarde, aumenta el desconocimiento que se tiene sobre la posición estimada por el filtro. Este comportamiento se puede observar en las gráficas de la figura C4-5, que han sido generadas para el ejemplo bSv (medidas originales y desordenadas estimadas por cada uno de los algoritmos sin etapa de validación). Las curvas del IFAsyn y Bayes de la gráfica (a) coinciden.

En la grafica superior se presenta el valor de la varianza de cada estado frente al tiempo, mientras que en las dos inferiores se representa el resultado del proceso de validación para el algoritmo Bayes e IFAsyn, y para el algoritmo de Kalman clásico. Para los algoritmos Bayes e IFAsyn todas las medidas son válidas (al no existir etapa de validación que pueda rechazar ninguna medida), tanto las que llegan sin retardo (V) como las que llegan retardadas (T&V). Sin embargo, para el algoritmo de Kalman, las medidas retardadas (MT) no serán consideradas válidas y no podrán ser incorporadas como evidencia. Para cada estado se puede observar que las etapas de crecimiento de la varianza aparecen en los periodos de falta de información sensorial asociado a dicho estado, o retardo en la recepción de la misma.

Por lo tanto, se comprueba que el filtro de Kalman tiene un comportamiento peor cuando hay medidas retardadas, al no ser capaz de utilizar la información sensorial asociada a las medidas que llegan fuera de tiempo, y que cuando esto no ocurre los tres algoritmos son equivalentes.

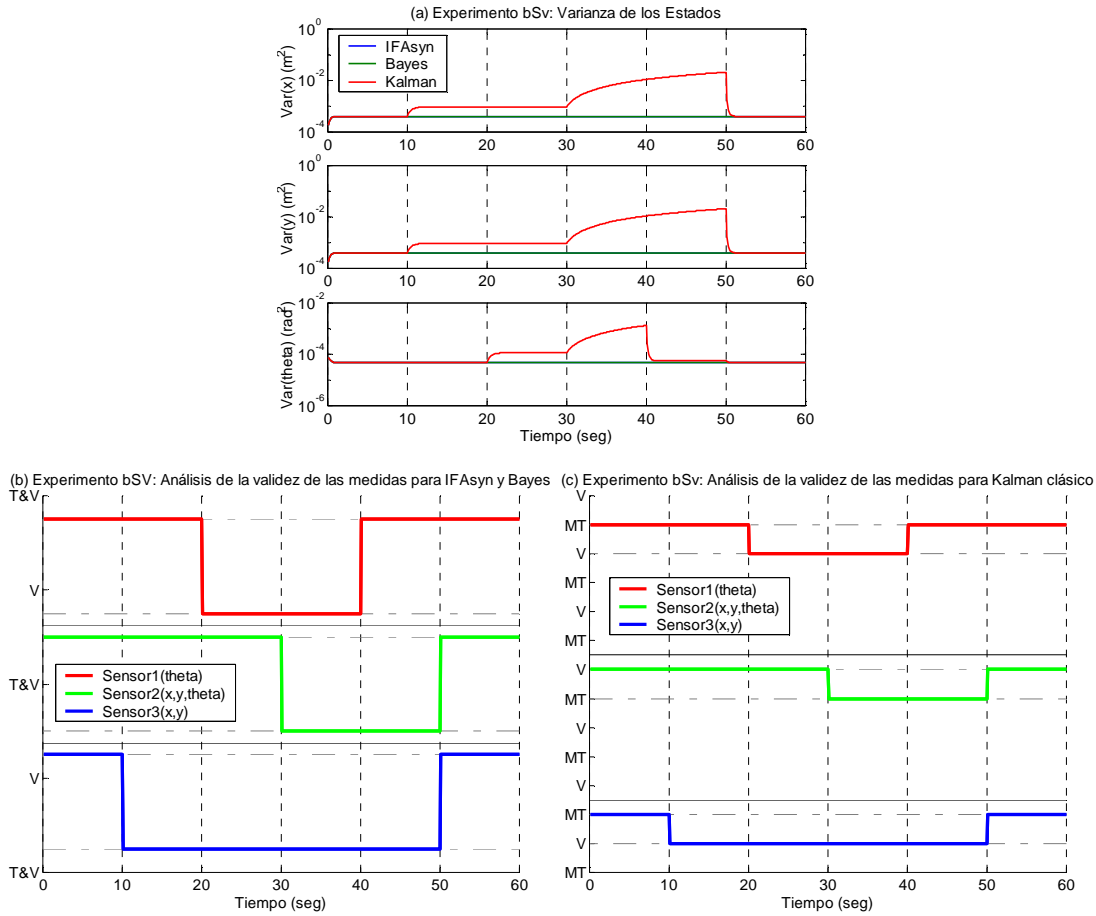


Figura C4-5. Varianza final del estado para Bayes, IFAsyn y Kalman para el ejemplo de medidas originales y retardadas, cuando no se utiliza la etapa de validación. En las gráficas inferiores se representa que medidas han sido procesadas por ser válidas (V válidas, T&V retardadas y válidas) y cuales han sido rechazadas por haber llegado fuera de tiempo (MT).

Influencia del desorden de las medidas en los resultados obtenidos

Con el objeto de demostrar la influencia que el desorden de las medidas introduce en cada uno de los algoritmos, se compararán los valores que los tres algoritmos (Bayes, IFAsyn y Kalman) obtienen de la estima del estado y su varianza en todos los instantes de muestreo, para parejas de ejemplos que únicamente se diferencian en el orden en el que las medidas son recibidas por los algoritmos. Además, también se comprobará si las mismas medidas han sido incorporadas como evidencia en los algoritmos o si han sido rechazadas, bien porque han llegado tarde y no han podido ser procesadas (en el caso de Kalman) o porque han sufrido una suerte diferente durante el proceso de validación.

Para realizar esta comparación, obtendremos para cada algoritmo la diferencia máxima existente entre los valores estimados del estado (y la varianza) entre los ejemplos que únicamente se diferencian por el orden o desorden de sus medidas. Para el análisis relativo a la suerte sufrida por las medidas, se contabilizará el número de las mismas que han sido tratadas de forma diferente. En

la siguiente tabla se recogen los resultados de este análisis, para cada pareja de experimentos similares, salvo por la existencia de medidas ordenadas o desordenadas. Ésta se encuentra dividida en dos partes, en la primera se recogen los resultados cuando algoritmos no realizan la etapa de validación mientras que en la segunda se presentan los resultados cuando dicha etapa es utilizada para detectar la existencia de medidas erróneas.

Experimento		IFAsyn: Orden vs. Desorden			Bayes: Orden vs. Desorden			Kalman: Orden vs. Desorden		
		Estado	Var.	Valid	Estado	Var.	Valid	Estado	Var.	Valid
Sv	a vs. b	0	0	0	2.1e-14	1.6e-19	0	1.4e-1	2.1e-2	800
	c vs. d	0	0	0	2.1e-14	1.6e-19	0	1.4e-1	2.1e-2	700
	e vs. f	0	0	0	3.5e-14	1.6e-19	0	8.1	2.0e-2	800
Cv	a vs. b	1.1e-2	2.5e-5	5	1.1e-2	2.5e-5	5	1.6e-1	2.1e-2	807
	c vs. d	1.1e-2	2.5e-5	5	1.1e-2	2.5e-5	5	1.6e-1	2.1e-2	707
	e vs. f	1.1e-2	2.5e-5	5	1.1e-2	2.5e-5	5	1.5e-1	2.0e-2	808

Tabla C4-3. Comparativa de los resultados obtenidos por el mismo algoritmo cuando las medidas se encuentran ordenadas y desordenadas, para las 6 combinaciones posibles de los 12 ejemplos. En la primera columna se representan los experimentos, agrupados en dos subconjuntos (Sv sin validación y Cv con validación) e identificados con la letra indicativa del experimento propuesto en la figura C4-2 a la que se refieren los resultados (*a* medidas originales ordenadas, *b* medidas originales desordenadas, *c* sensor no mide y medidas ordenadas, *d* sensor no mide y medidas desordenadas, *e* medidas erróneas y ordenadas, *f* medidas erróneas y desordenadas). Para cada algoritmo se representa la diferencia máxima existente entre el valor del estado estimado cuando las medidas se encuentran ordenadas y desordenadas, la diferencia máxima existente entre los valores de la varianza, y el número de medidas que han sido tratadas de forma diferente por cada algoritmo. En verde se resalta, para cada pareja de experimentos, el algoritmo que obtiene los resultados más parecidos, mientras que en naranja se muestran los casos cuya diferencia no puede considerarse despreciable.

De las comparaciones realizadas para cada algoritmo, los resultados obtenidos por Kalman para las 6 parejas de experimentos son los que tienen un mayor grado de disparidad, fenómeno esperable, ya que en los ejemplos con medidas ordenadas el filtro de Kalman es capaz de utilizar todas las medidas (no erróneas, cuando haya etapa de validación) mientras que en los ejemplos con medidas desordenadas, las medidas que lleguen temporalmente retardadas al algoritmo (800 para los ejemplos *a*, *b*, *e* y *f*, y 700 para los ejemplos *c* y *d*) son desechadas. La discrepancia máxima de 8 (metros) en los estados para la comparativa de eSv vs. fSv es debida a que se trata de dos ejemplos con medidas erróneas sin validación, y en el caso en el que las medidas no se encuentran retardadas las medidas erróneas confunden al algoritmo de Kalman antes que en el caso con retardo (donde las medidas erróneas retardadas son desechadas debido al retardo).

Para los algoritmos de Bayes e IFAsyn es importante distinguir los resultados obtenidos cuando se realiza la etapa de validación (experimentos Cv) y cuando esta etapa es obviada (experimentos Sv).

En los experimentos sin etapa de validación (Sv), los dos algoritmos obtienen resultados equivalentes para el caso ordenado y desordenado. La discrepancia existente en el caso de Bayes toma valores despreciables, debidos a los errores de redondeo que se introducen cuando al repropagar los estados se suma la nueva información y se resta la antigua. En el caso de IFAsyn la

discrepancia es nula, debido a que este algoritmo almacena por separado la información sensorial asociada a un instante de muestreo y toda la información asociada a instantes de muestreo anteriores, por lo que se evitan operaciones de suma y resta, y por lo tanto no se acumulan errores de redondeo en las mismas. Por lo tanto, el filtro de Información propuesto en el algoritmo IFAsyn presenta una estabilidad computacional mayor que el del algoritmo de inferencia exacto de Bayes. Alguno de los algoritmos para la asimilación de medidas desordenadas analizados en el capítulo anterior ([Nettleton2001], el AI de [ZhangK2003]), que ahorran memoria a costa de realizar operaciones equivalentes a los del algoritmo de inferencia exacto, presentarán el mismo problema de acumulación de errores de redondeo que el algoritmo de Bayes.

En los experimentos con validación (Cv), aparece una discrepancia máxima significativa en los resultados obtenidos por cada uno de los algoritmos entre el caso ordenado y desordenado. La razón de esta discrepancia se encuentra en la influencia que tiene en la etapa de validación el orden en el que se introducen las medidas. A la hora de desarrollar los algoritmos de Bayes e IFAsyn se decidió que el proceso de validación únicamente se realizaría una vez por cada medida, tan pronto como ésta fuese recibida y procesada por el algoritmo. Por lo tanto, la información asociada a las medidas tomadas en el instante t_{m1} que llegan temporalmente retardadas al algoritmo en un instante t_{r1} , no influirá en el proceso de validación de medidas que han sido tomadas en instantes t_{m2} , posteriores a t_{m1} y anteriores a t_{r1} , y recibidas por el algoritmo en cualquier t_{r2} anterior a t_{r1} .

Analizando los resultados de la columna relativa a las discrepancias en la validación (*Valid*) de la tabla C4-3 para cada pareja de ejemplos propuestos para los algoritmos de Bayes e IFAsyn con etapa de validación (Cv), se puede observar que 5 de las medidas han sufrido una suerte diferente durante el proceso de validación. Es decir, 5 medidas han sido validadas/rechazadas para el ejemplo ordenado mientras que 5 medidas han sido rechazadas/validadas para su equivalente desordenado. Como comprobaremos a continuación, estas diferencias en la etapa de validación son la causa de la discrepancia máxima existente en los estados y la varianza.

Para poder ilustrar mejor las consecuencias de la influencia de la etapa de validación en los resultados obtenidos con medidas ordenadas y desordenadas, analizaremos con más detalle, a lo largo de todos los instantes de muestreo, los resultados y discrepancias de una pareja de ejemplos. En particular, los relativos a la pareja *aCv* vs. *bCv*, cuyos resultados se encuentran representados en las gráficas de la figura C4-6.

En las dos gráficas superiores, (a) y (b), se representa el resultado de la etapa de validación, para cada uno de los ejemplos, sensores e instante de muestreo. En las mismas se distinguen si la medida ha llegado a los algoritmos sin retardo y ha sido validada (V) o rechazada (NV), o si la medida ha llegado retardada y ha sido validada (T&V) o rechazada (T&NV). Se puede observar

que una parte de las medidas proporcionadas por cada sensor es rechazada debido al intervalo de confianza elegido para el test chi-cuadrado (95%).

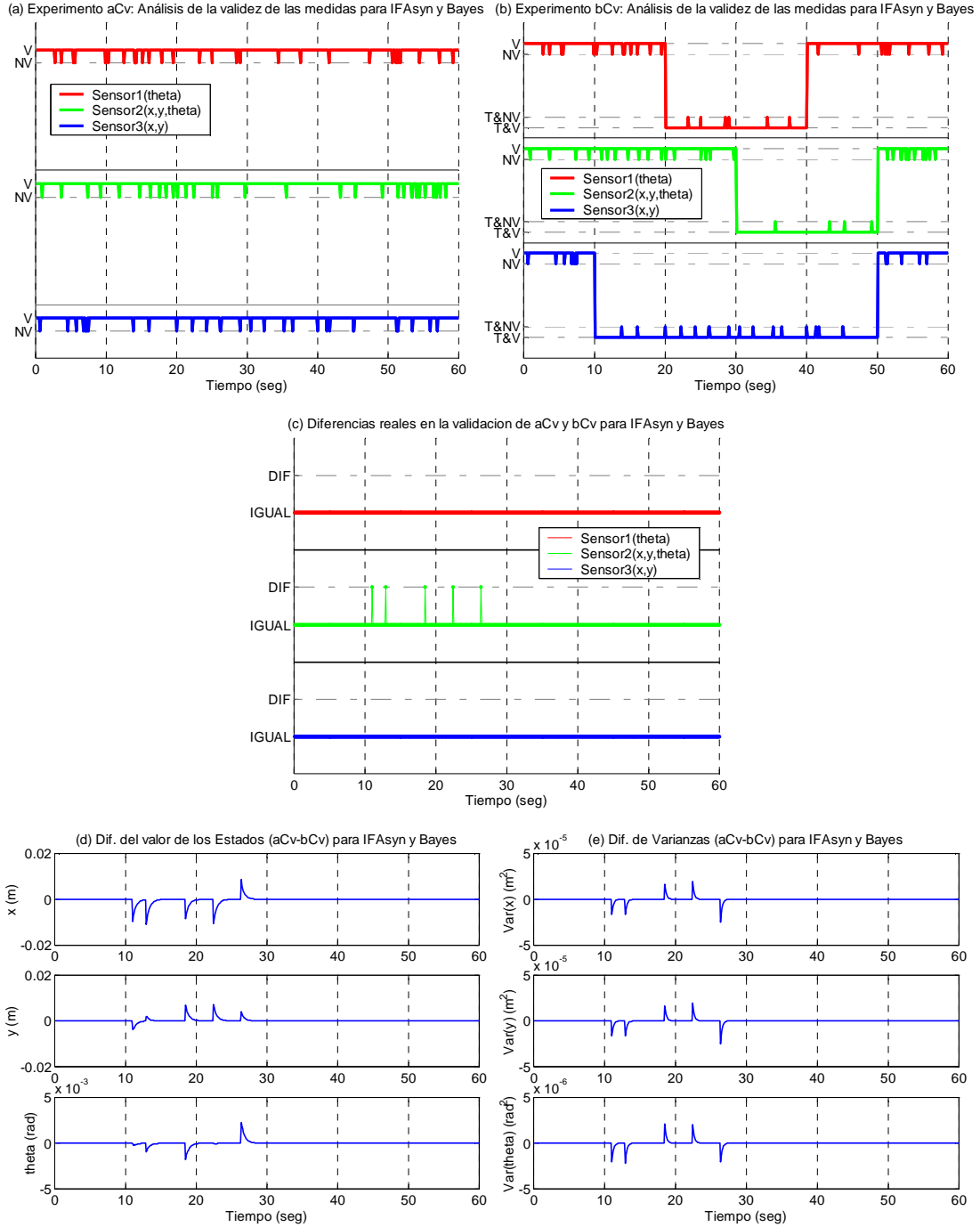


Figura C4-6. Discrepancia existente entre los ejemplos aCv (medidas originales ordenadas, algoritmo con etapa de validación) y bCv (medidas originales desordenadas, algoritmo con etapa de validación) para los algoritmos de Bayes e IFAsyn, debida a la diferente suerte sufrida por 5 medidas durante la etapa de validación. En las gráficas (a) y (b) se representa la suerte sufrida por las medidas durante la validación (V válidas, NV no válidas, T&V tarde y válida, T&NV tarde y no válida) y en (c) las diferencias (DIF) en el proceso de validación entre los ejemplos aCv y bCv (considerando iguales las que llegando ordenadas/desordenadas han sido validadas o no en ambos casos). En (d) y (e) se representa la diferencia existente para los estados y las varianzas entre ambos experimentos.

En la gráfica intermedia (c), se representa si las medidas han sufrido la misma suerte durante el proceso de validación (IGUAL) o no (DIF), sin distinguir entre el caso ordenado y el retardado. En ella se puede ver que las 5 medidas que han sufrido diferente suerte son medidas proporcionadas por el sensor2, y que la discrepancia se da en instantes de muestreo en los que hay medidas retardadas del sensor1 o del sensor3. Por lo tanto, se comprueba que los retardos de los otros dos sensores influyen en el proceso de validación del sensor2.

Por último, en las dos gráficas inferiores se representa la diferencia entre los valores estimados por el estado (d) y de la varianza (e) para los dos experimentos. Es sencillo comprobar que los picos de discrepancia en ambos casos se dan en los instantes de muestreo cercanos a la discrepancia en la validación. Por lo tanto, la influencia del retardo en la validación se observará principalmente en el entorno de los puntos donde el proceso de validación da diferentes valores para el caso ordenado y retardo. Si el número de puntos en lo que esto ocurre es pequeño, como en el ejemplo mostrado, este fenómeno puede considerarse despreciable.

Además, es importante destacar que la discrepancia existente entre los valores estimados por los algoritmos de Bayes e IFAsyn entre el estado y la varianza para medidas ordenadas y desordenadas, a pesar de existir, es pequeña y significativamente menor (en ordenes de magnitud) que la que existe en el caso del filtro de Kalman donde las medidas retardadas son rechazadas. Y no solo es menor, sino que se da únicamente de forma puntual, mientras que en el filtro de Kalman aumentará hasta que éste vuelva a ser capaz de procesar medidas (no retardadas). Para destacar más esta diferencia, en la figura C4-7 se representan, para el filtro de Kalman, las graficas correspondientes a las de la figura C4-6. Como se puede ver en la gráfica (c) de la figura C4-7, el número de medidas de cada sensor que sufren una suerte distinta durante el proceso de validación ha aumentado de forma considerable, ya que para el caso desordenado, a las medidas no validadas que llegan a tiempo, hay que sumarle todas aquellas que no son válidas porque han llegado tarde. De hecho, las coincidencias (IGUAL de la gráfica (c)) durante los instantes de muestreo en el que las medidas se encontraban retardadas (MT de la gráfica (b)) son debidas a que en el caso ordenado esas medidas fueron rechazadas por la etapa de validación. Durante los instantes de muestreo en los que las discrepancias existen debido al rechazo de medidas retardadas, aparecen discrepancias significativas en el valor estimado del estado (gráfica (d)) y la varianza (gráfica (e)). En el caso de la varianza, la discrepancia se hace especialmente significativa cuando los sensores retardados no introducen información relativa a cada estado: del segundo 30 al 50 para las medidas de posición x e y , y del segundo 30 al 40 para las medidas de orientación θ . Esto es debido a que la falta de información sensorial hace que la varianza se vea únicamente modificada por la etapa de predicción y por lo tanto, aumente. El crecimiento de la discrepancia entre los instantes en los que no hay información sensorial, también se manifiesta en la gráfica de la discrepancia de los estados.

Por último es importante llamar la atención del lector que desee comparar las gráficas (d) y (e) de las figuras C4-6 y C4-7, sobre el hecho de que los ejes de ordenadas tienen escalas diferentes.

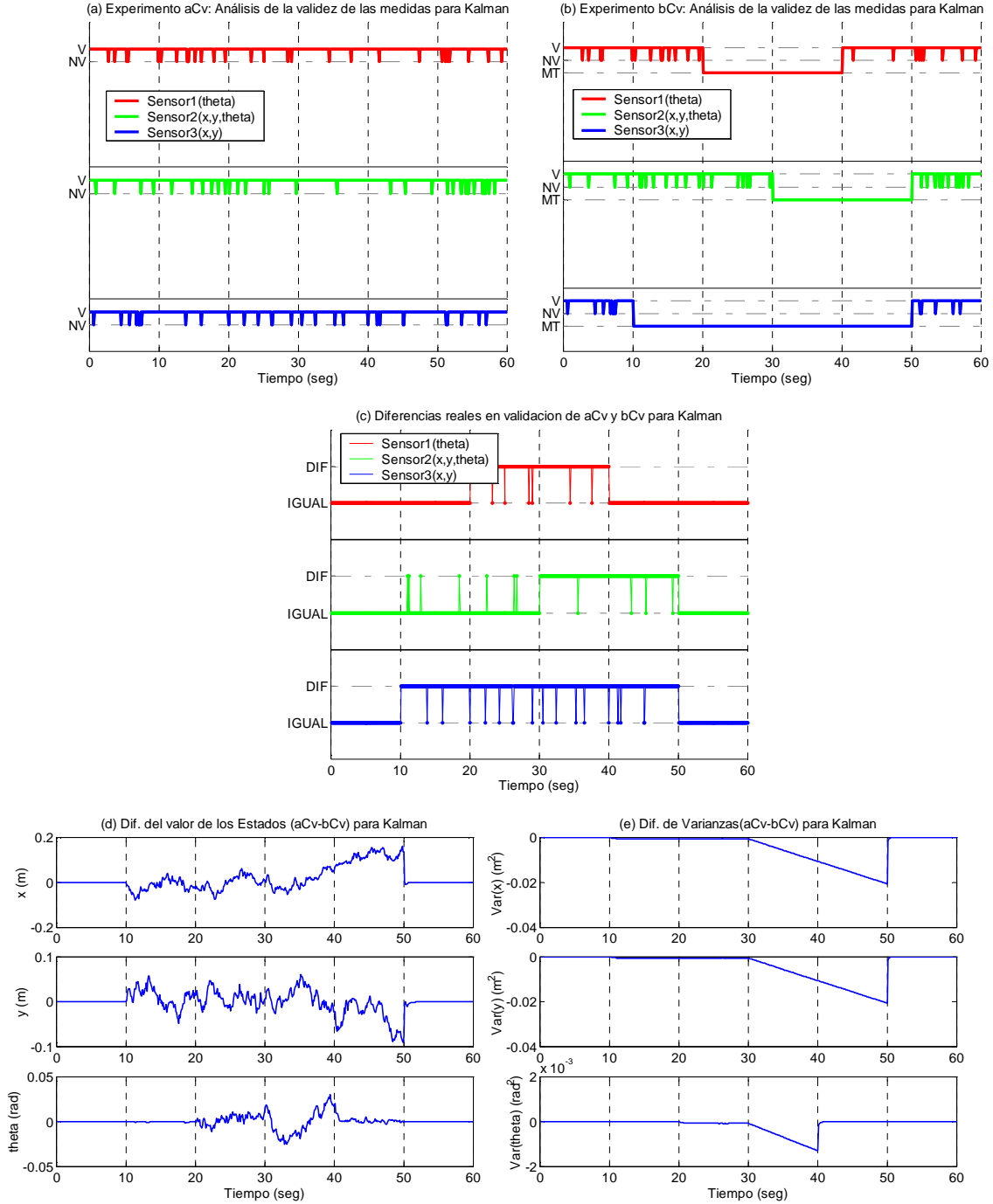


Figura C4-7. Discrepancia existente entre los ejemplos aCv (medidas originales ordenadas, algoritmo con etapa de validación) y bCv (medidas originales desordenadas, algoritmo con etapa de validación) para el filtro de Kalman (que desecha las medidas retardadas). En las gráficas (a) y (b) se representa la suerte sufrida por las medidas de cada ejemplo durante la validación (V válidas, NV no válidas, MT medidas desechadas por llegar tarde), en (c) las diferencias (DIF) en el proceso de validación entre ambos ejemplos (considerando iguales aquellas que han sido aceptadas/rechazadas en los dos ejemplos), y en (d) y (e) la diferencia existente para los estados y las varianzas entre ambos ejemplos.

Por lo tanto, se ha vuelto a comprobar que el comportamiento del filtro de Kalman ante las medidas retardadas, es peor que el de los algoritmos de Bayes e IFAsyn, ya que estos son capaces de procesar la información retardada que Kalman desecha; se ha observado que el algoritmo IFAsyn es computacionalmente más estable que el algoritmo de Bayes (y que parte de los algoritmos para la asimilación de medidas temporalmente desordenadas analizados en el capítulo anterior que ahorran memoria a costa de realizar operaciones equivalentes al algoritmo de Bayes), ya que el primero almacena la información sensorial de un instante y de los instantes previos en variables diferentes mientras que el segundo realiza sumas y restas para actualizarla; y se ha analizado la influencia del desorden de las medidas en el proceso de validación, que afecta puntualmente a los resultados obtenidos por ambos algoritmos en los instantes en los que una revalidación de las medidas cambiaría el resultado de dicho proceso.

Importancia de la existencia de la etapa de validación cuando hay medidas erróneas.

Al igual que ocurre en el filtro de Kalman, la existencia de medidas erróneas debe de ser tratada por los algoritmos de Bayes e IFAsyn, bien modelando el error de las medidas en la matriz de varianza, o bien implementando una etapa de validación que permita rechazar las medidas erróneas, y por lo tanto evite asociar la información sensorial asociada a las mismas. Por lo tanto, en los dos algoritmos propuestos en esta tesis para sistemas lineales, se ha incluido una etapa de validación, que se puede deshabilitar cuando no sea necesaria, y que permitirá rechazar las medidas no válidas cuando éstas existan.

Para ilustrar la importancia de su existencia, analizaremos, a través de las gráficas de la figura C4-8, los resultados obtenidos por el algoritmo IFAsyn para el ejemplo f (medidas erróneas y desordenadas) cuando éste es ejecutado con y sin etapa de validación. El análisis con el algoritmo de Bayes es equivalente, ya que los resultados de ambos filtros son similares, salvo por errores despreciables de redondeo.

En el ejemplo elegido, no sólo se suministran a IFAsyn medidas realmente erróneas, sino que parte de las mismas llegan al filtro de forma temporalmente desordenada, lo que nos permitirá comprobar que el algoritmo funciona correctamente ante ambos factores cuando se incluye una etapa de validación. Las medidas erróneas del sensor3 han sido obtenidas a partir de las medidas básicas generadas, modificándolas a partir de los 30 segundos con un desplazamiento de 10 metros.

En las dos primeras gráficas de la figura C4-8 se representa la suerte que cada medida ha sufrido durante la validación para fSv y fCv . En la gráfica (a) todas las medidas son consideradas válidas ya que el algoritmo no realiza la etapa de validación. En la gráfica (b) aparecen medidas erróneas puntuales en todos los sensores, porque durante su generación aleatoria algunas se han salido del intervalo de confianza elegido para el test de validación chi-cuadrado, y todas las medidas

realizadas por el sensor3 a partir de los 30 segundos son rechazadas, porque el algoritmo ha sido capaz de detectar que son erróneas.

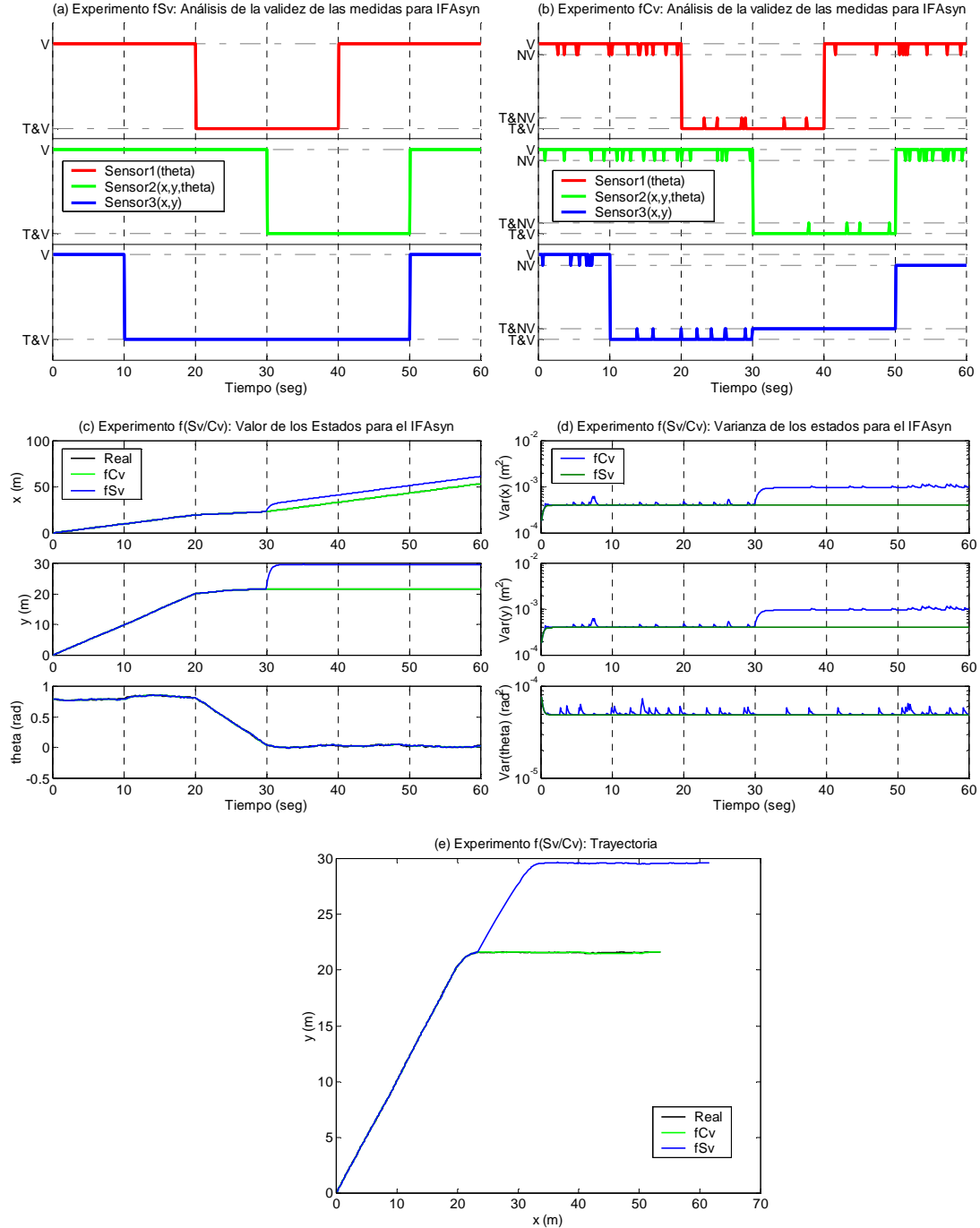


Figura C4-8. Análisis del comportamiento de IFAsyn cuando se incorporan medidas erróneas, y el algoritmo se ejecuta sin y con etapa de validación. En las gráfica (a) y (b) se representa la suerte sufrida, para los casos fSv y fCv respectivamente, por cada una de las medidas durante la validación (V válidas, NV no válidas, T&V válidas y retardadas, T&NV no válidas y retardadas), en (c) y (d) los valores de los estados y de la varianza para cada ejemplo, y en (e) la trayectoria del móvil.

En la gráfica (c) la figura C4-8 se representa el valor del estado estimado por IFAsyn, cuando éste es ejecutado con y sin etapa de validación. A partir de los 30 segundos, instante en el que las medidas del sensor3 son erróneas, los resultados obtenidos para los estados x e y para el ejemplo sin validación (fSv) se desvían significativamente de los obtenidos con validación (fCv), mientras éstos últimos son similares a los valores reales generados en la simulación. Esto es debido a que las medidas erróneas del sensor3, relacionadas únicamente con los estados x e y , engañan al filtro sin etapa de validación, al no ser éste capaz de detectar que el valor de las mismas se aleja significativamente del valor de medida esperado. La gráfica (e) la figura C4-8 también recoge, bajo el formato de trayectoria, la discrepancia existente entre los valores estimados de x e y .

En la gráfica (d) la figura C4-8 se representa el valor de la varianza asociada a cada estado para ambos ejemplos. La discrepancia en la varianza existe de forma puntual para los tres estados, ya que en el ejemplo con validación (fCv) se rechazan algunas medidas de los tres sensores a lo largo de todos los instantes de muestreo, y de forma continua y significativa en los dos primeros estados a partir de los 30 segundos, instante en el que el sensor3 empieza a proporcionar información errónea sobre la posiciones x e y . El valor es siempre mayor para el ejemplo con validación, debido a que al rechazarse medidas, la información utilizada para estimar la posición del móvil es menor, y por lo tanto el desconocimiento (medido en la varianza) que se tiene sobre la posición es mayor. Es decir, el filtro con validación, cuando rechaza una medida, estima un valor con una precisión menor, mientras que el filtro sin validación, utiliza todas las medidas, válidas y erróneas, y a la estima obtenida le asocia una varianza pequeña, que indica que cree que el valor obtenido es mejor.

Por lo tanto, las medidas erróneas “engañan” al algoritmo sin validación en dos aspectos: una, haciendo que estime un valor incorrecto a partir de información errónea y otra, haciendo que además considere que su estima es mejor que en el caso en el que no utilizase dicha información incorrecta. Este comportamiento erróneo del filtro de Información asíncrono para medidas retardadas (IFAsyn) sin validación, que también ocurre en el algoritmo de inferencia bayesiana (Bayes) y en el filtro de Kalman (cuando no hay retardos), justifica la necesidad de una etapa de validación, para los casos en los que existan medidas cuyo error no se encuentre modelado en la matriz de varianza del sensor. La necesidad se da, tanto para el caso en el que las medidas se reciban temporalmente ordenadas (único caso soportado por el filtro de Kalman), como para el caso contrario (soportado también por los algoritmos de Bayes e IFAsyn).

Evolución del valor del estado y varianza para cada instante de muestreo según se va asimilando la información medida en instantes anteriores y que llega al algoritmo temporalmente retardada

Otro aspecto que se puede analizar es la evolución, para cada instante de muestreo, del valor estimado del estado por los algoritmos de Bayes e IFAsyn, según se va incorporando a los mismos

la información que llega temporalmente retardada. Como los dos algoritmos son equivalentes, únicamente se realizará el análisis para IFAsyn.

Los ejemplos elegidos para este análisis son el de medidas originales y desordenadas (b) y el del sensor que no mide y sus medidas están desordenadas (d). A continuación se analizan los resultados obtenidos para los dos ejemplos cuando el algoritmo usa o no usa la etapa de validación.

En las gráficas de la figura C4-9 se representa el valor de la varianza de los estados para tres instantes diferentes del proceso de asimilación, cada uno representado por una curva en un color diferente.

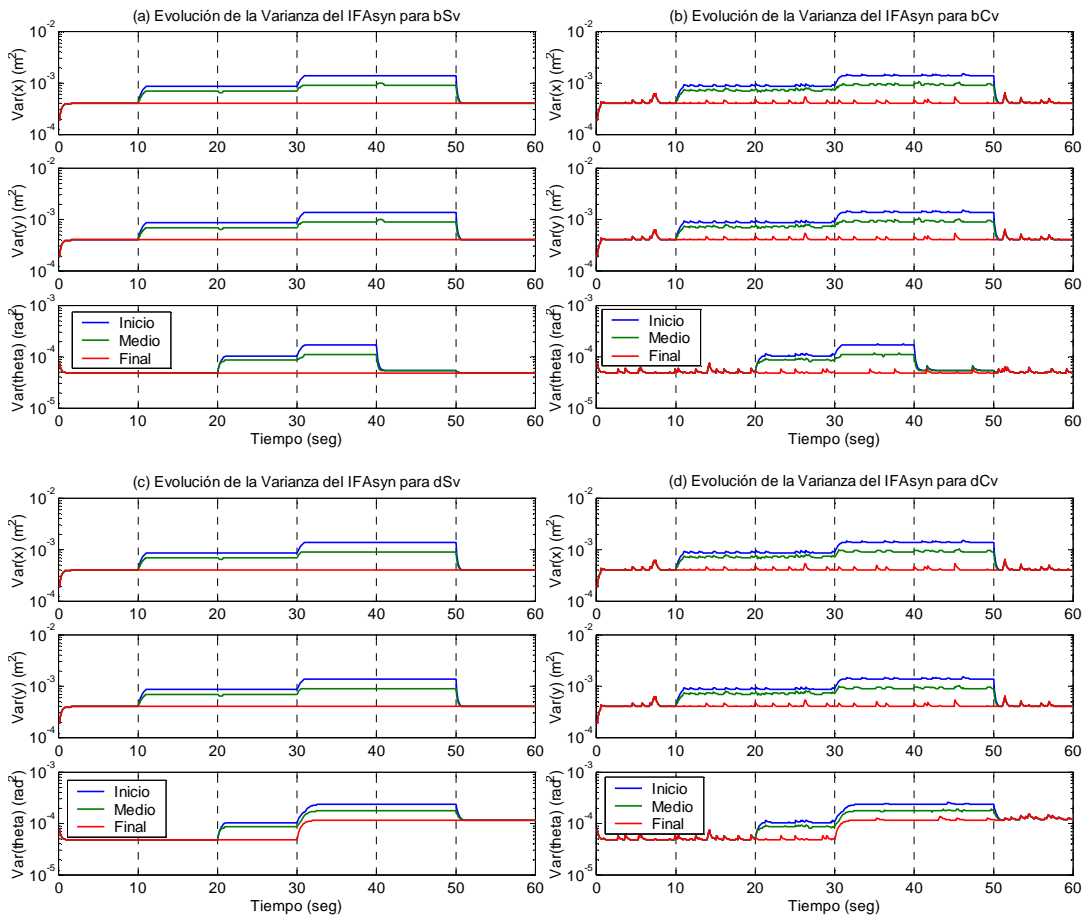


Figura C4-9. Evolución del valor de la varianza para distintos experimentos, según IFAsyn va asimilando la información retardada. La curva identificada como Inicio corresponde a la varianza del estado que IFAsyn calcula con la información sensorial disponible hasta el correspondiente instante de muestreo. La curva Final representa el valor de la varianza una vez que se ha asimilado toda la información retardada. En la curva Medio se dibuja el valor de la varianza cuando se ha asimilado, para el instante de muestreo correspondiente, la mitad de la información sensorial que ha sido tomada en instantes anteriores y es asimilada en instantes posteriores.

Para cada instante de muestreo t , se representa la varianza calculada por el algoritmo en dicho instante t , con diferente información sensorial:

- La varianza de la curva *Inicio* (azul) ha sido calculada con la información sensorial que ha sido medida y recibida por el algoritmo hasta el instante t .
- La varianza de la curva *Final* (rojo) ha sido calculada con toda la información sensorial que ha sido medida hasta el instante t , es decir una vez que se ha asimilado toda la información retardada medida hasta t .
- La varianza de la curva *Medio* (verde) ha sido calculada tras la asimilación de la mitad de la información medida hasta t que se recibe en instantes posteriores a t .

Para los cuatro ejemplos se puede observar como la varianza disminuye según se va asimilando la información temporalmente desordenada, ya que los valores de curva *Inicial* son mayores o iguales a los de la curva *Medio*, y a su vez éstos últimos están por encima de los de la curva *Final*. En los casos con validación (bCv y dCv) las curvas de varianza presentan aumentos puntuales de la varianza, debido al rechazo de medidas consideradas no válidas por estar fuera del intervalo de confianza elegido para el test de validación chi-cuadrado. Salvo por esos aumentos puntuales, las curvas de las parejas de experimentos representados en la misma fila de la figura, que se diferencian por la existencia o no de la validación, son semejantes.

Para los casos del experimento con medidas originales y retardadas (bSv y bCv), se puede observar que las curvas *Iniciales* de la varianza de los estados x e y sufren dos aumentos significativos, uno a partir del segundo 10 y otro a partir del segundo 30, debido al comienzo de la zona de información retardada para el sensor3 y el sensor2. Estos aumentos se producen durante 1 segundo, y posteriormente los valores de las curvas se estabilizan, ya que empieza a llegar al sistema la información que tiene un retardo de 1 segundo. A partir de los 50 segundos, el valor de ambas curvas decrece rápidamente, debido a que se está asimilando la información retardada anterior a los 50 segundos y la información ordenada posterior a dicho instante. En las curvas *Iniciales* correspondientes a la varianza del estado θ se observa un comportamiento análogo, aunque relativo a los instantes en que los sensores empiezan y dejan de proporcionar la información temporalmente desordenada. Como en este caso el fin de las zonas temporalmente desordenadas del sensor1 y el sensor2 no coincide, el decremento de la varianza de θ de la curva *Inicial* se realiza en dos etapas, aunque el segundo es difícil de apreciar a simple vista.

Para los casos del experimento donde el sensor1 deja de proporcionar información sensorial a partir de los 30 segundos (dSv y dCv), las curvas asociadas a los estados x e y son similares a la de los ejemplos anteriores, debido a que dicho sensor no proporciona información asociada a los mismos. Sin embargo, las curvas de varianza del estado θ presentan un comportamiento diferente, ya que a partir de los 30 segundos, el sensor deja de funcionar. Por lo tanto, la caída de la varianza asociada al instante 40 desaparece, y solo se produce la asociada al instante 50.

El análisis de la evolución del valor de los estados no es tan directo, ya que según se incluye nueva información sensorial su valor puede ser modificado en una dirección o en la contraria, de acuerdo con la información sensorial que se este asimilando en cada momento. Lo que es importante destacar es que si los modelos del sistema y de los sensores son correctos, el valor del estado se parecerá habitualmente, según se va asimilando más información retardada, cada vez más a su valor real. En la figura C4-10 se representa la diferencia entre el valor estimado del estado y el real, para los ejemplos de bSv y bCv, en los tres instantes de asimilación anteriormente expuestos (*Inicial*, *Medio* y *Final*). En ella se puede observar que en los instantes en los que hay una discrepancia significativa, la diferencia con el valor final (curva roja) suele ser menor que el resto. Los instantes en que las diferencias coinciden son aquéllos en los que no hay información sensorial retardada.

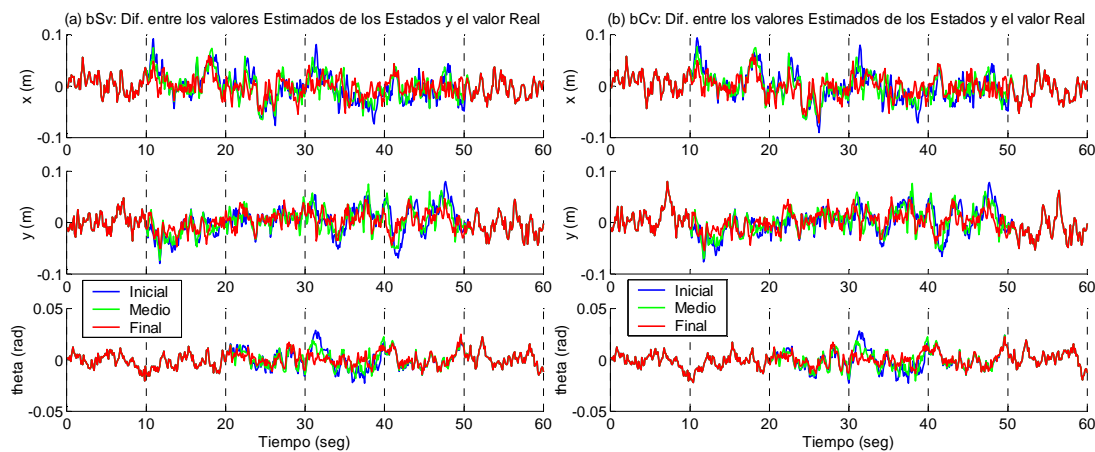


Figura C4-10. Evolución de la diferencia entre el valor estimado del estado y el valor real para los experimentos bSv y bCv, según los algoritmos van asimilando la información retardada. La curva identificada como Inicio corresponde a la diferencia cuando el estado es calculado con la información sensorial disponible hasta el correspondiente instante de muestreo. La curva Final representa el valor de la diferencia una vez que se ha asimilado toda la información retardada. En la curva Medio se dibuja el valor de la diferencia cuando se ha asimilado, para el instante de muestreo correspondiente, la mitad de la información sensorial que ha sido tomada en instantes anteriores y es asimilada en instantes posteriores.

Por lo tanto, se ha observado como para cada instante de muestreo t , la varianza del estado disminuye según se va asimilando la información sensorial retardada medida hasta dicho instante, y que el valor del estado, no sólo se conoce con mayor exactitud (al tener una varianza menor), sino que habitualmente converge hacia su valor real.

▪ Análisis del consumo de memoria y de las necesidades de computo (grupos experimentales I y II)

Otros aspectos a tener en cuenta en la comparación de los tres algoritmos (Bayes, IFAsyn y Kalman) son las necesidades de memoria y el número de operaciones realizadas en cada uno de los casos. Los valores representativos necesarios para este tipo de análisis se obtendrán de medir, sobre

la versión de los algoritmos implementados sobre Matlab, el tamaño de los vectores y matrices utilizadas para almacenar las variables utilizadas por cada algoritmo y el número de operaciones en punto flotante realizadas durante la ejecución de los mismos.

Antes de analizar los resultados obtenidos es importante tener en cuenta que:

- A la hora de calcular la memoria únicamente se mide aquella que es estrictamente necesaria para las matrices, vectores y escalares que almacenan las variables propuestas en los algoritmos de Bayes, IFAsyn y Kalman. La memoria adicional utilizada para facilitar la implementación de las operaciones u organizar las variables en estructuras y vectores no ha sido contabilizada.
- En el cálculo de las operaciones, sólo se contabilizan aquéllas en punto flotante que *Matlab* devuelve como el número de operaciones realizadas cuando se ejecuta el comando *flops*. Por lo tanto, hay que tener en cuenta que algunas de las operaciones, como son las comparaciones del test de validación, no serán contabilizadas. Aun así, el número de operaciones calculado por *Matlab* nos sirve para realizar un estudio del número de operaciones realizadas para cada ejemplo por cada algoritmo, ya que en todos los casos se contabilizará el mismo tipo de operaciones, y por lo tanto, los valores obtenidos serán comparables.

Para comparar el número de operaciones y memoria utilizada se utilizarán los ejemplos propuestos en los grupos experimentales I y II. En las dos secciones siguientes se presentan y analizan los resultados obtenidos.

Grupo Experimental I: Influencia en los algoritmos de la existencia de retardos y errores en el funcionamiento y medidas de los sensores

En las gráficas de la siguiente figura se representa, para cada uno de los 12 ejemplos (6x2 posibilidades de validación) de este grupo experimental, el número de operaciones en punto flotante (flops) y la cantidad de memoria utilizada para almacenar las variables necesarias para el correcto funcionamiento de los algoritmos. Los experimentos, cuyo identificador se encuentra representado en el eje de abscisas, se han ordenado en dos subconjuntos, el primero formado por los experimentos con medidas ordenadas y el segundo por aquellos realizados con medidas desordenadas, ya que las necesidades de memoria y de cálculo son significativamente diferentes entre los elementos de cada grupo.

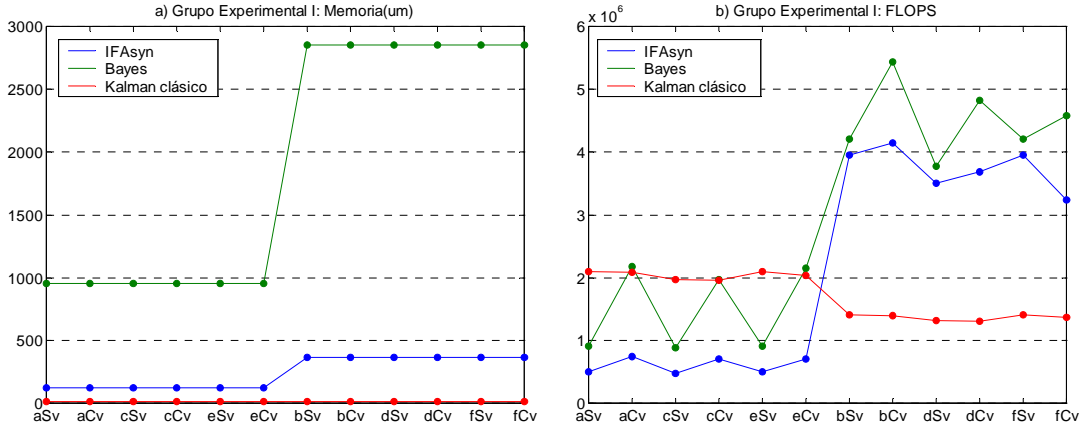


Figura C4-11. Unidades de memoria y número de operaciones en punto flotante, para los tres algoritmos para los 12 (6x2 posibilidades de validación) ejemplos propuestos en el grupo experimental I. Cada experimento se encuentra identificado con tres letras, la primera indica a cual de los experimentos propuestos en la figura C4-2 se refieren los resultados (*a* medidas originales ordenadas, *b* medidas originales desordenadas, *c* sensor no mide y medidas ordenadas, *d* sensor no mide y medidas desordenadas, *e* medidas erróneas y ordenadas, *f* medidas erróneas y desordenadas) y las dos siguientes si el algoritmo se ha ejecutado sin etapa de validación (Sv) o con ella (Cv).

La memoria utilizada por el filtro de Kalman clásico es despreciable (solo hay que almacenar el valor del estado y la varianza asociada al último instante de muestreo) frente a la memoria utilizada por los otros dos algoritmos y su valor no depende (al permanecer constante) del experimento realizado. Para los algoritmos de Bayes e IFAsyn las necesidades de memoria cambian al pasar de los experimentos con medidas ordenadas a experimentos con medidas desordenadas, debido a que es necesario incrementar el tamaño de la ventana de tiempo del algoritmo para poder almacenar la información necesaria para la asimilación de las medidas temporalmente desordenadas. Para el caso ordenado, se eligió una ventana de tiempo de 0.5 segundos, mientras que para el caso desordenado se le asignó un valor de 1.5 segundos. Se puede comprobar que la memoria necesaria para el algoritmo IFAsyn es significativamente menor que la utilizada por el algoritmo de Bayes, ya que para el algoritmo IFAsyn para cada instante de muestreo sólo se almacenan las variables de predicción ($\hat{\mathbf{y}}_{t|AASIM \leq t-1}$ e $\mathbf{Y}_{t|AASIM \leq t-1}$), de información sensorial (\mathbf{i}_t e \mathbf{I}_t) y de la señal de control (\mathbf{u}_{t-1}); mientras que para el caso del algoritmo de Bayes hay que almacenar los potenciales de los conglomerados y separadores ($\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t)$, $\Phi_{\mathbf{x}_t\mathbf{z}_{s,t}}(\mathbf{x}_t, \mathbf{z}_{s,t})$, $\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}(\mathbf{x}_t)$ y $\Phi_{\mathbf{x}_t\mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_t)$).

El número de operaciones, para los ejemplos con medidas ordenadas, es menor para el algoritmo IFAsyn que para el filtro de Kalman, debido a que los cálculos necesarios para la etapa de asimilación multisensorial son más sencillos en el filtro de Información que en el de Kalman (el porqué de esta diferencia se encuentra explicado en el apéndice I). Lo mismo ocurre para Bayes respecto a Kalman cuando no se realiza la etapa de validación, ya que durante el diseño del algoritmo de Bayes se ha reducido al máximo el número de operaciones para asegurar su correcto funcionamiento con el menor tiempo de cálculo.

Para los ejemplos con medidas desordenadas, el número de operaciones menor le corresponde a Kalman debido a que desecha toda la información que le llega temporalmente retardada. En cambio, los algoritmos de Bayes e IFAsyn realizan un número significativamente mayor de operaciones con el objeto de asimilar la información retardada y predecir el valor de dicha información hasta el último instante de muestreo.

El algoritmo de Bayes realiza en todos los casos más operaciones que el algoritmo IFAsyn, debido a que las operaciones sobre potenciales son más complejas, al almacenar ésta información sobre probabilidad conjunta y no únicamente sobre el estado y varianza del estado del sistema. La diferencia se reduce para los ejemplos sin validación (*Sv), ya que para dicho caso, en el algoritmo de Bayes únicamente se realizan las absorciones necesarias para propagar la información de los nodos sensoriales a los de estado. En cambio, en los ejemplos con validación (*Cv) la diferencia aumenta ya que para realizar el test chi-cuadrado hay que realizar absorciones en el sentido opuesto.

Por lo tanto, se ha comprobado que el algoritmo IFAsyn es mejor, desde un punto de vista computacional y de memoria, que el algoritmo de Bayes. Además, para el caso de tener la información ordenada y múltiples sensores, la complejidad operacional asociada al filtro de Información es menor que la asociada al filtro de Kalman multisensorial.

El consumo de memoria menor siempre se da para el filtro de Kalman, ya que dicho algoritmo solo almacena el valor del estado y su varianza para el último instante de muestreo. Como contrapartida a dicho ahorro de memoria, no es capaz de asimilar la información que llega al algoritmo de forma temporalmente retardada. Una implementación clásica del filtro de Información, no válida para el caso temporalmente desordenado, tendrá un gasto computacional menor que Kalman, y un consumo de memoria similar a la del filtro de Kalman, ya que en ese caso no sería necesario almacenar en variables diferentes la información sensorial del instante t (en \mathbf{i}_t e \mathbf{I}_t) y la información del estado calculada tras la predicción ($\hat{\mathbf{y}}_{t|AASIM \leq t-1}$ e $\mathbf{Y}_{t|AASIM \leq t-1}$).

Grupo Experimental II: Influencia del retardo en el consumo de memoria y número de operaciones

En los ejemplos del grupo experimental I el retardo elegido toma sólo dos valores diferentes (0 y 1 segundos), hecho que no permite analizar con detalle la influencia de la duración del retardo en el consumo de memoria y el número de operaciones. Por lo tanto, para llevar a cabo el estudio de dicha influencia, utilizaremos los ejemplos del grupo experimental II, cuyas medidas tienen un retardo variable R de 0 a 8 segundos. Antes de comenzar con el análisis, es importante destacar que para asegurar que las medidas retardadas pueden ser asimiladas por los algoritmos de Bayes e IFAsyn y minimizar, en la medida de lo posible, el consumo de memoria para todos los casos, la ventana de tiempo con la que se han ejecutado los algoritmos se ha ajustado al retardo de las

medidas de cada ejemplo. En la siguiente figura se presentan las gráficas que recogen la memoria utilizada y el número de *flops* realizadas por los tres algoritmos con etapa de validación, durante la ejecución de los 18 (9 con retardo variable únicamente en un sensor y 9 con retardo variable en los tres sensores) ejemplos de este grupo experimental.

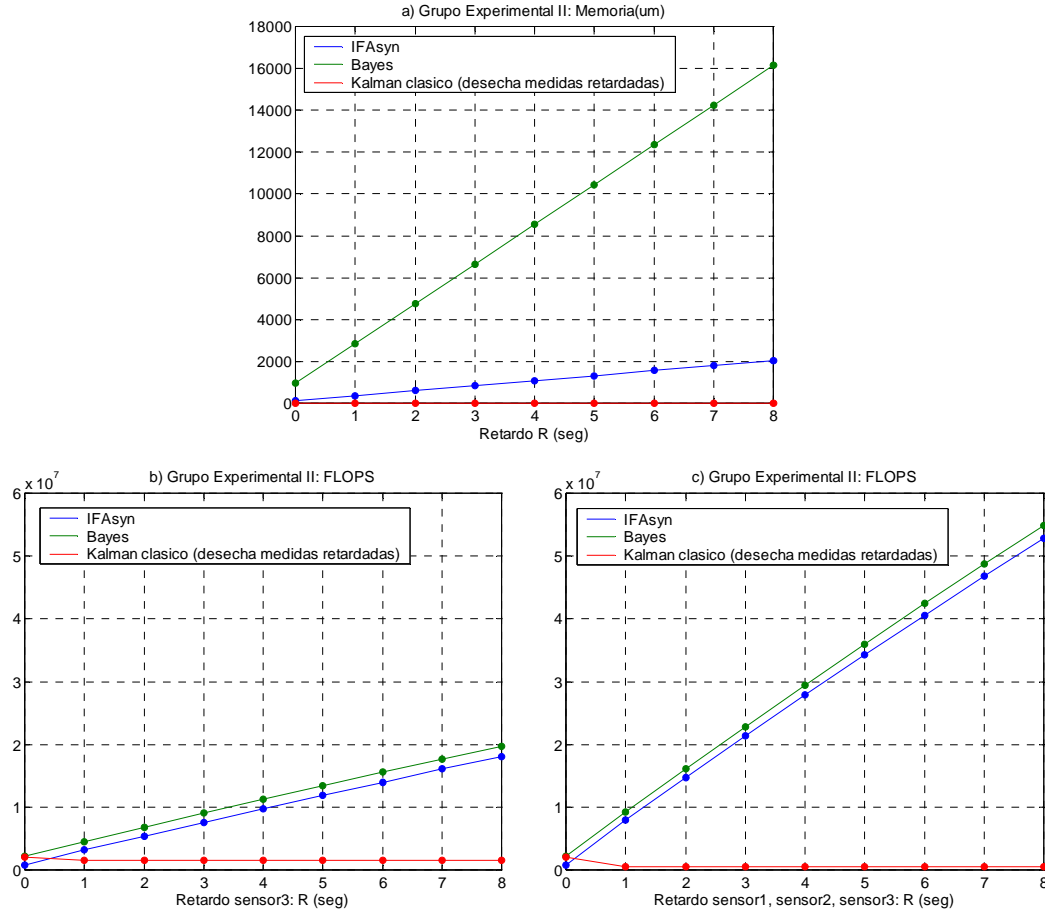


Figura C4-12. Unidades de memoria y número de operaciones en punto flotante, para los tres algoritmos para los 18 (9 retardos x 2 número de sensores retardado) ejemplos propuestos en el grupo experimental II.

En la gráfica superior se representan las unidades de memoria utilizadas para los 9 retardos en los dos subconjuntos, mientras que en las gráficas inferiores se presenta el número de operaciones en punto flotante contabilizadas por Matlab realizadas para los 9 retardos y, el subconjunto de experimentos con un único sensor retardado (izquierda) y el subconjunto con los tres sensores retardados (derecha).

La memoria utilizada por los algoritmos de Bayes e IFAsyn no depende del número de sensores cuyas medidas se encuentran temporalmente retardadas, por lo que el consumo de memoria es el mismo para parejas de ejemplos con el mismo retardo R y distinto número de sensores retardados. En la gráfica superior de la figura C4-12 se representa, para cada uno de los 9 valores de retardo (desde 0 a 8 segundos, en incrementos de 1 segundo), las unidades de memoria utilizadas por los 3 algoritmos. El mejor de los casos, cuyo valor además se mantiene constante, se presenta para el filtro de Kalman clásico, debido a que este almacena únicamente el estado y su varianza para el último instante de muestreo. El algoritmo IFAsyn presenta un comportamiento peor, que se

incrementa linealmente con el aumento del retardo en las medidas, y por lo tanto del valor de la ventana de tiempo, porque para cada instante de muestreo t dentro de dicha ventana es necesario almacenar las variables \mathbf{i}_t , \mathbf{I}_t , $\hat{\mathbf{y}}_{t|AASIM \leq t-1}$, $\mathbf{Y}_{t|AASIM \leq t-1}$ y \mathbf{u}_{t-1} . El peor de los casos se presenta para el algoritmo de Bayes, ya que en este caso, para cada instante de muestreo t hay que guardar los potenciales de los conglomerados de transición $\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t)$ y de medidas $\Phi_{\mathbf{x}_t\mathbf{z}_{s,t}}(\mathbf{x}_t, \mathbf{z}_{s,t})$, y los de sus separadores correspondientes ($\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}(\mathbf{x}_t)$ y $\Phi_{\mathbf{x}_t\mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_t)$). La pendiente de la curva asociada al algoritmo de Bayes es significativamente mayor que la del algoritmo IFAsyn, por lo que para sistemas con sensores con retardos significativos, el consumo de memoria asociado a Bayes será mucho peor.

El número de operaciones realizadas por los algoritmos de Bayes e IFAsyn para cada uno de los 18 ejemplos es diferente, ya que por cada medida retardada hay que realizar una asimilación de dicha medida (para Bayes una absorción desde el nodo de transición al de medida y otra en el sentido contrario) y un conjunto de predicciones (absorciones de izquierda a derecha entre nodos de transición de estados) sucesivas hasta alcanzar el último instante de muestreo. Por lo tanto, si hay una medida retardada, este conjunto de operaciones se realizará una vez, mientras que si hay N medidas retardadas, se realizará N veces. Este comportamiento se puede observar comparando las curvas de las gráficas inferiores de la figura C4-12 asociadas al mismo algoritmo (Bayes o IFAsyn). La pendiente de la curva cuando existe un único sensor con medidas retardadas es significativamente menor que la pendiente de la curva cuando las medidas de los tres sensores tienen retardo.

El comportamiento del filtro de Kalman, cuando se analiza el número de operaciones realizadas, es justamente el contrario: el número de operaciones realizadas cuando los tres sensores tienen medidas retardadas es menor que para el caso en el que sólo hay un sensor con retardo. Esto se debe a que todas las medidas retardadas se desechan, y por lo tanto, un aumento en el número de medidas que llegarán tarde implica una disminución en el número de operaciones realizadas. Cuando los tres sensores se encuentran retardados, las únicas operaciones contabilizadas son aquellas asociadas con la predicción, ya que toda la información sensorial llega tarde y por lo tanto es desechada por el filtro de Kalman.

Dentro del mismo subconjunto experimental (formado por 9 experimentos con distinto retardo) se puede observar que el número de operaciones realizadas por el filtro de Kalman clásico, salvo para el caso en el que el retardo es nulo, siempre es menor que el número de operaciones realizadas por los otros dos algoritmos, debido a la eliminación de las medidas retardadas. Para el mismo retardo, IFAsyn siempre realiza menos operaciones que Bayes porque la asimilación, validación y predicción de IFAsyn es menos costosa computacionalmente que las operaciones de absorción de Bayes. En los experimentos sin retardo, el algoritmo que realiza menos operaciones es IFAsyn,

debido a que la asimilación de la información procedente de múltiples sensores es menos compleja cuando se realiza en el espacio de información que en el espacio de estados.

Por lo tanto, se ha comprobado que cuando existe información retardada y para poder asimilarla es necesario utilizar los algoritmos de Bayes e IFAsyn, y que el algoritmo IFAsyn presenta un consumo de memoria menor y realiza operaciones menos costosas que el algoritmo de Bayes.

C4.2. Análisis del funcionamiento de los algoritmos para sistemas con modelos no lineales

En esta sección se estudian distintos aspectos del funcionamiento del algoritmo C3-3, el filtro de Información asíncrono para sistemas mixtos (lineales y/o no lineales) con sensores cuyas medidas pueden ser proporcionadas de forma retardada y temporalmente desordenada. Este algoritmo, que identificaremos como IFAsynMixto, también se encuentra recogido, a modo de resumen, en el apéndice II.

Para realizar el estudio de esta sección también se utilizará el algoritmo de Kalman Extendido habitual para sistemas no lineales (cuyas etapas se encuentran propuestas en la tabla A1-10 del apéndice I), que desechará las medidas que le lleguen de forma temporalmente retardadas, y una versión modificada del algoritmo C3-3, identificada y recogida en el apéndice II bajo el nombre de IFAsynMixtoEsp, para analizar la importancia del recálculo de la información sensorial asociada a la medida de los sensores no lineales.

Durante los experimentos de este estudio se usarán distintos sistemas (lineales y no lineales) con el objeto de poner de manifiesto distintos aspectos del funcionamiento de los algoritmos. Lo común a todos los sistemas utilizados es que su vector de estado está formado por tres componentes, que serán observadas, de diferentes formas, por tres sensores del sistema. Al igual que en la sección del análisis del funcionamiento de los algoritmos para sistemas con modelos lineales, las expresiones del modelo del sistema y sensorial elegido servirán para: (1) generar, mediante simulación y a partir de un valor del estado inicial y los valores de la señal de control, las medidas de los sensores; y (2) obtener, a partir de dicho estado inicial, de las medidas de los sensores y de las señales de control, el valor estimado del estado por los diferentes algoritmos.

Se examinará el comportamiento del algoritmo IFAsynMixto bajo diferentes condiciones experimentales, como son el consumo de memoria y número de operaciones necesarias para el funcionamiento ante diferentes retardos en la recepción y número de ecuaciones de modelo no lineales, la influencia que tiene el termino corrector de la medida en la asimilación de la información sensorial, la influencia del recálculo de la información asociada a las medidas no lineales debido a la existencia de información sensorial retardada, etc.

La organización de este estudio es diferente a la del realizado sobre los sistemas con modelos puramente lineales, porque se utilizarán distintos sistemas para ilustrar los distintos aspectos analizados, y, por lo tanto, resulta más conveniente presentar el sistema, los datos experimentales y los resultados de cada modelo de forma consecutiva. Además, se comenzará el análisis por el punto último de la sección anterior: el consumo de memoria y el número de operaciones realizadas, y se terminará el análisis con el estudio de aspectos propios del algoritmo C3-3. Esta inversión del orden en el que se estudian los aspectos del algoritmo IFAsynMixto es debida a que para el análisis de la memoria y operaciones utilizaremos el sistema lineal de la sección anterior, dándole un tratamiento no lineal, y por lo tanto, para ese estudio no es necesario introducir un nuevo modelo del sistema.

Tras estas consideraciones, únicamente queda presentar la organización de este apartado, que se encuentra dividido en dos secciones. En la primera se analizará la influencia que tiene en las necesidades de memoria y número de operaciones realizadas la existencia de retardo y de sensores con modelos no lineales. En la segunda se estudiarán otros aspectos relevantes, como la necesidad del término corrector de medida en la asimilación de la información sensorial, la influencia en los resultados obtenidos del recálculo de la información sensorial asociada a los sensores no lineales, etc.

C4.2.1. Análisis del consumo de memoria y de las necesidades de cómputo

Dos de los aspectos fundamentales del algoritmo IFAsynMixto son la memoria que hay que reservar para almacenar todas las variables que aseguren un correcto funcionamiento del mismo, y la sobrecarga computacional asociada a la asimilación de la información sensorial temporalmente retardada y a la reasimilación de la información sensorial asociada a los sensores no lineales. La memoria y el número de operaciones dependen directamente del retardo de recepción de las medidas y del número de sensores no lineales que tenga el sistema. Esta dependencia será mostrada a través de los ejemplos propuestos, en los que se modificará el valor del retardo entre 0 y 8 segundos, con una diferencia de 1 segundo entre dos valores sucesivos, y el número de sensores no lineales existentes en el sistema.

Para realizar el análisis se utilizarán el sistema lineal propuesto en la sección C4.1.1 y los datos experimentales básicos presentados en la figura C4-1 de misma sección. La linealidad del sistema elegido no sólo no es un impedimento para ejecutar sobre el mismo el algoritmo IFAsynMixto (ya que los sistemas puramente lineales son tan sólo un caso particular sobre los que el algoritmo puede estimar el estado) sino que además, nos permitirá, indicándole al algoritmo que parte de los sensores lineales no lo son, medir el incremento de memoria y la sobrecarga computacional asociadas directamente a la reasimilación, debida a la recepción de medidas temporalmente desordenada, de las medidas de los sensores que el algoritmo considera que son no lineales. Al

utilizar para todos los experimentos de esta sección el mismo modelo sensorial, aunque sea siempre lineal, se podrá realizar una comparación directa de la memoria necesaria y del número de operaciones realizadas, ya que el tamaño de los vectores de medida y el número de operaciones realizadas cada vez que se ejecutan las funciones de medida (ecuación de modelo y obtención del jacobiano) se mantendrá constante. En el estudio propuesto no se considera la linealidad o no linealidad de la ecuación de transición de estados, ya que la sobrecarga computacional será siempre la misma ante ambas suposiciones.

Sobre los datos experimentales básicos de la sección C4.1.1 se constituirá el grupo experimental III, formado por un total de 72 experimentos, que se pueden dividir (según el número de sensores no lineales, de 0 a 3) en 4 subconjuntos de 18 experimentos. A su vez cada subconjunto de 18 experimentos se puede dividir (de acuerdo con el número de sensores retardados, 1 –el sensor3- o los 3- sensor1, sensor2 y sensor3-) en dos subconjuntos de 9 elementos (donde la variable que diferencia a los elementos de cada subconjuntos es el valor del retardo R de los sensores que proporcionan medidas retardadas). Esta clasificación de los ejemplos del grupo experimental III se encuentra esquematizada en las gráficas de la siguiente figura.

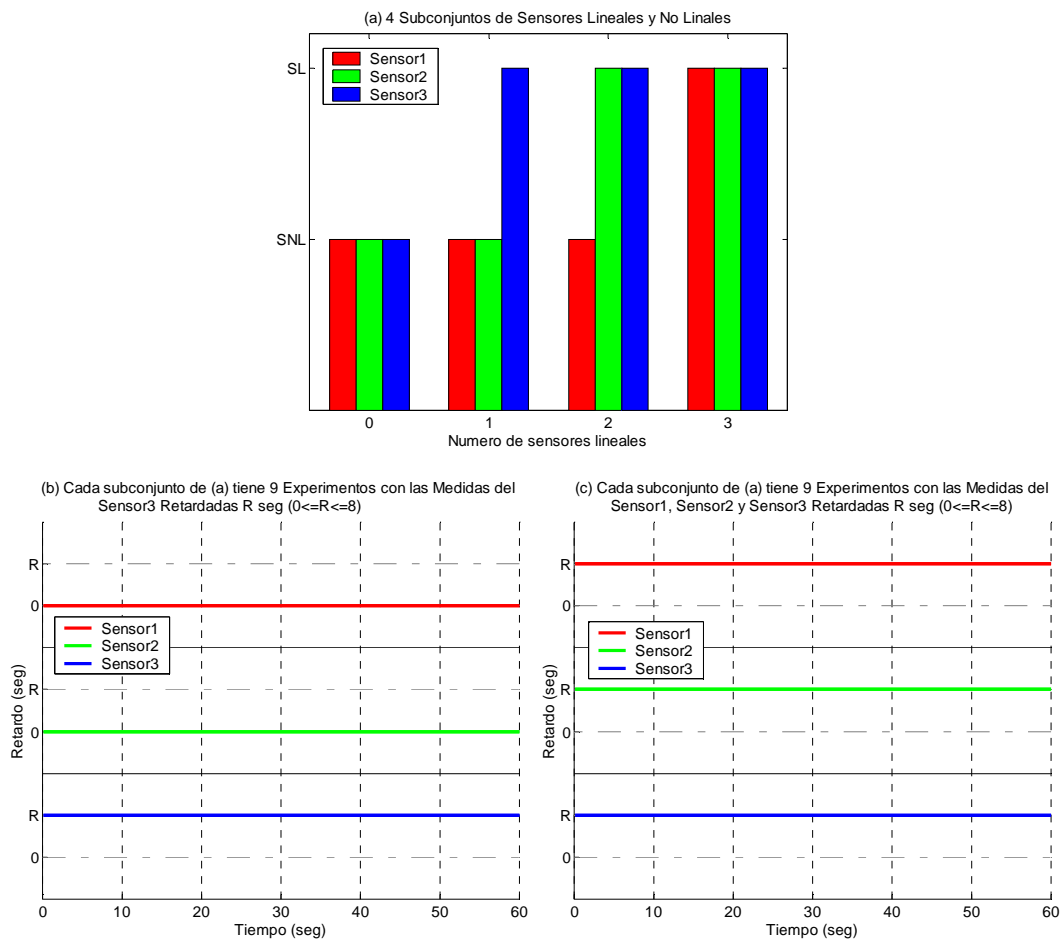


Figura C4-13. Caracterización de los experimentos del Grupo Experimental III: Influencia de la duración del retardo y el número de sensores no lineales en el consumo de memoria y número de operaciones.

En la gráfica (a) se muestra, para cada uno de los 4 subconjuntos de 18 experimentos, que sensores de los mismos son lineales y cuales no, mientras que en las otras dos gráficas se representa, para cada grupo de 18 experimentos, el retardo de cada uno de los sensores durante todo el experimento para los dos subconjuntos de 9 elementos (cada uno de los cuales tomar un valor del retardo \mathbf{R} en el intervalo de 0 a 8 segundos). Siguiendo la notación utilizada en las gráficas, a lo largo de esta sección nos refriéremos a los experimentos con un único sensor retardado como experimentos tipo (b), mientras que los experimentos con los tres sensores retardados serán denominados experimentos tipo (c).

Para realizar el análisis de la influencia del número de sensores no lineales y del retardo en las necesidades de memoria y el número de operaciones del algoritmo IFAsynMixto éste se ha ejecutado, con la etapa de validación, sobre todos y cada uno de los experimentos del grupo experimental III, caracterizado en la figura C4-13. Además, para el caso del subconjunto experimental con todos los sensores lineales, también se ejecutará el algoritmo de Bayes con etapa de validación. De esta forma podremos comparar si la sobrecarga computacional y el aumento de necesidades de memoria asociados a la falta de linealidad de los sensores para el algoritmo IFAsynMixto hace que la medida del consumo de memoria y del número de operaciones supere a las realizadas por el algoritmo de Bayes, cuyo comportamiento en esos dos aspectos era significativamente peor que el del algoritmo IFAsyn, versión puramente lineal del algoritmo IFAsynMixto.

Las necesidades de memoria y las operaciones realizadas se han medido siguiendo las mismas directrices que las utilizadas durante la sección del análisis del consumo de memoria y de las necesidades de cómputo del apartado del análisis de funcionamiento de los algoritmos para sistemas con modelos lineales.

Las necesidades de memoria de los algoritmos de Bayes e IFAsynMixto no dependen del número de sensores que tengan retardo, sino únicamente del valor del retardo elegido y del número de sensores no lineales. Por lo tanto, las gráficas de las necesidades de memoria, que se encuentran recogidas en la figura C4-14, no distinguen entre el tipo de experimentos utilizado, (b) o (c), del grupo experimental III. En la gráfica (a) se puede observar como la memoria mínima necesaria para almacenar los potenciales de los conglomerados y separadores del algoritmo de Bayes, que solo puede ser utilizado para el caso de sistemas puramente lineales y por lo tanto para el subconjunto de ejemplos con 3 sensores lineales, es siempre peor que las necesidades mínimas de memoria del algoritmo IFAsynMixto, que además de almacenar la información sensorial y de predicción en parejas de valores diferentes, y el valor de la señal de control, tiene que recoger por cada sensor no lineal que ha medido los valores de las medidas asimiladas. La diferencia entre el algoritmo de Bayes e IFAsynMixto crece significativamente con el aumento del retardo de las medidas. En la gráfica (b) se observa como el consumo de memoria en el algoritmo IFAsynMixto aumenta según

se incrementa el número de sensores no lineales del sistema. Este comportamiento es debido a que cada medida $\zeta_{s,t1}$ recibida de un sensor no lineal tiene que ser almacenada en una variable, para permitir al algoritmo volver a calcular la información sensorial asociada a la misma en el caso en que el algoritmo reciba posteriormente cualquier otra medida $\zeta_{s,t2}$ que ha sido realizada con anterioridad ($t2 < t1$).

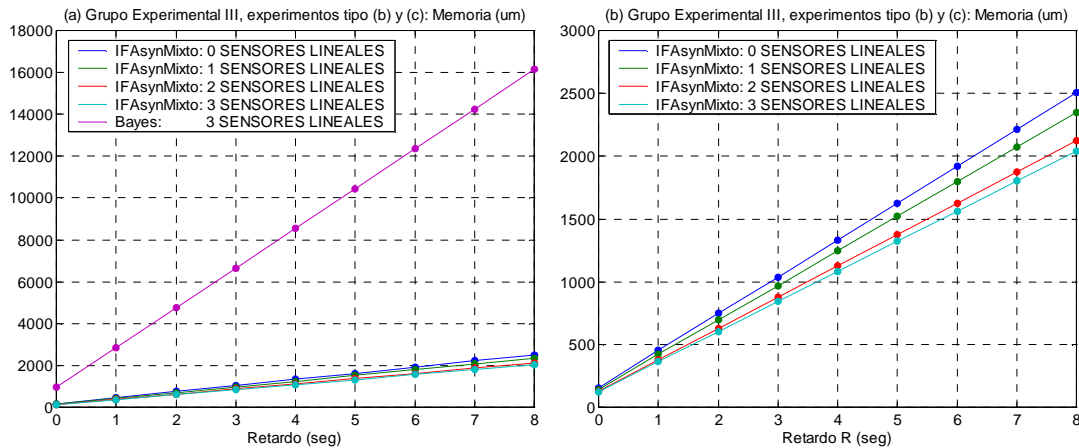


Figura C4-14. Análisis de la memoria utilizada por el algoritmo IFAsynMixto y Bayes, ambos con etapa de validación, para los experimentos del grupo experimental III, sobre los que se puede comprobar la influencia sobre la memoria del retardo existente en la recepción de las medidas y el número de sensores no lineales. El consumo de memoria es el mismo para los experimentos tipo (b) y (c). Las dos gráficas son similares, salvo por la escala del eje de ordenadas y que en la primera se incluye la curva que representa la memoria utilizada por el algoritmo de Bayes mientras que en la segunda (b) sólo se observa la memoria necesaria para las pruebas del algoritmo IFAsynMixto.

El número de operaciones realizadas es diferente para los experimentos tipo (b), donde únicamente hay un sensor retardado, que para los experimento tipo (c), donde los tres sensores se encuentran retardados. Esta diferencia se puede observar, respectivamente, en las gráficas (a) y (b) de la figura C4-15.

En la gráfica (a) de la figura C4-15 se puede observar como el número de operaciones realizadas por el algoritmo IFAsynMixto aumenta con el incremento del valor del retardo R del sensor3 y la disminución del número de sensores lineales. Esto es debido a que, por una parte, un incremento de R significa tener que realizar más operaciones para repropagar la información sensorial retardada hasta el último instante de muestreo, y una disminución en el número de sensores lineales, tener que realizar más operaciones de recálculo de la información sensorial asociada a los sensores no lineales. Las curvas del algoritmo IFAsynMixto para los casos de 1 y 0 sensores lineales son casi coincidentes, debido a que la diferencia entre esos dos grupos experimentales reside únicamente en el paso del sensor3 de lineal a no lineal, y dicho sensor es el que proporciona las medidas retardadas. Por lo tanto, no será necesario recalcular la información asociada a dicho sensor, ya que no llega ninguna medida desordenada con respecto al mismo. Sin embargo, el caso 0 será ligeramente peor, ya que para calcular la información sensorial asociada a las medidas del sensor3

se utilizará la expresión no lineal, que requiere más operaciones al incluir el término corrector de la medida. El algoritmo de Bayes, que se puede utilizar únicamente para el caso puramente lineal, es computacionalmente más costoso para el caso lineal del algoritmo IFAsynMixto. Para el resto de los subconjuntos experimentales, el algoritmo que realiza menos operaciones depende del valor del retardo R del sensor3: para retardos pequeños es mejor el algoritmo IFAsynMixto, mientras que para valores elevados será mejor el algoritmo de Bayes, a pesar de no poderse ejecutar sobre los ejemplos con sensores no lineales.

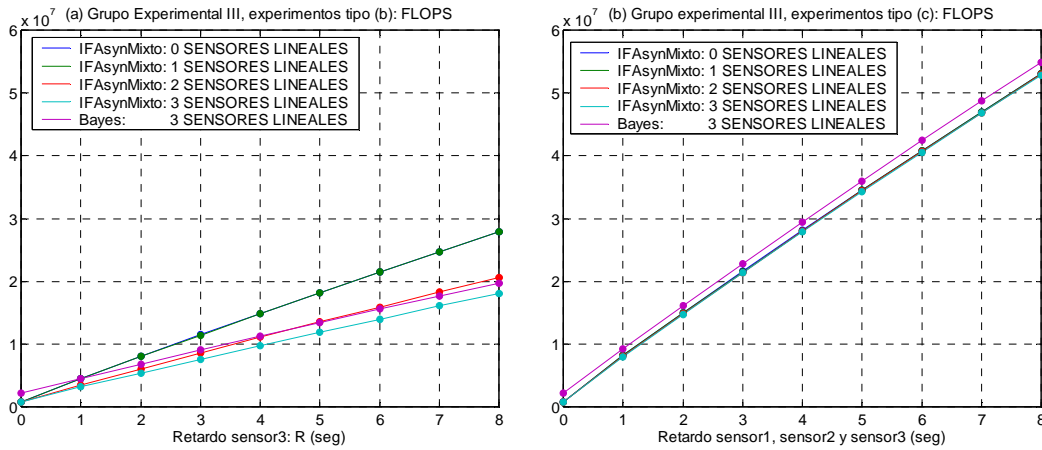


Figura C4-15. Análisis del número de operaciones realizadas por el algoritmo IFAsynMixto y Bayes, ambos con etapa de validación, para los experimentos del grupo experimental III, sobre los que se puede comprobar la influencia sobre la memoria del retardo existente en la recepción de las medidas y el número de sensores no lineales. En la gráfica (a) se representa el número de operaciones realizadas para los experimentos tipo (b), es decir con un único sensor retardado, mientras que en la gráfica (b) se representa el número de operaciones realizadas para los experimentos tipo (c), es decir, con los tres sensores retardados.

En la gráfica (b) de la figura C4-15 se puede observar el número de operaciones realizadas por los algoritmos Bayes e IFAsynMixto, ambos con validación, cuando los tres sensores proporcionan medidas retardadas, para diferentes valores del retardo R y número de sensores lineales. Las necesidades de cálculo son mayores que para los ejemplos de la gráfica (a), ya que el retardo en tres sensores en vez de uno incrementa, por cada sensor retardado, el número de operaciones que son necesarias para propagar la información sensorial retardada hasta el último instante de muestreo. Las curvas obtenidas para el algoritmo IFAsynMixto son similares para los cuatro subconjuntos de experimentos, ya que al encontrarse toda la información retardada el mismo tiempo, no es necesario recalculiar la información sensorial ya asimilada anteriormente que corresponde a instantes de muestreo posteriores. La diferencia entre las cuatro curvas, poco apreciable en la gráfica, se debe únicamente a que según aumenta el número de sensores no lineales (o se decrementa el número de sensores lineales) la asimilación de cada medida, realizada únicamente cuando la medida llega al algoritmo, es más costosa, porque los sensores no lineales utilizan en dicha operación el término corrector de la medida. Para cada valor del retardo R , el número de operaciones realizadas por el algoritmo de Bayes para los ejemplos lineales de los

experimentos tipo (c) es mayor que el número de operaciones realizadas por el algoritmo IFAsynMixto para los ejemplos lineales, mixtos y no lineales de los experimentos del mismo grupo.

Por lo tanto, se ha comprobado como para el algoritmo IFAsynMixto el número de operaciones y la memoria necesaria aumenta según se incrementa el valor del retardo y el número de sensores no lineales, utilizando para todos los ejemplos de la comparativa el mismo sistema puramente lineal, que el algoritmo tratará como un sistema no lineal y mixto, según los casos. La ventaja de esta elección es obvia: una modificación del sistema para cada ejemplo complicaría la realización del estudio, ya que el número de operaciones contabilizadas y la memoria necesaria pasaría también a depender del ejemplo elegido. El algoritmo de Bayes, que solo puede ser utilizado en el caso puramente lineal, ha sido incorporado a esta comparativa para mostrar que las necesidades de memoria del mismo son superiores en cualquier caso a las necesidades del algoritmo IFAsynMixto y las necesidades de computo del mismo son peores para el caso puramente lineal y para los casos no lineales con retardos pequeños.

C4.2.2. Análisis de los resultados obtenidos por el algoritmo C3-3 (IFAsynMixto) sobre distintos sistemas no lineales

En esta sección se analizan los resultados obtenidos, para dos sistemas mixtos (con ecuaciones lineales y no lineales) por el filtro de Información Mixto asíncrono para medidas desordenadas (IFAsynMixto). Los sistemas elegidos servirán para poner de manifiesto distintos aspectos de la influencia del desorden en las medidas en los resultados obtenidos y de la importancia del término corrector en los sensores no lineales.

▪ Análisis de los resultados para un sistema con funciones no lineales suaves

El primero de los sistemas, cuyas expresiones se encuentran recogidas en la tabla C4-4, modela un sistema con dos sensores lineales (sensor1 y sensor2) y con una ecuación de transición de estados y un sensor (sensor3) no lineales. La ecuación de transición de estado puede ser utilizado para modelar el comportamiento de un robot circular con dos ruedas motrices, separadas una distancia b , que pueden ser giradas, de forma independiente, de acuerdo con la señal que controla el espacio lineal recorrido por cada una de las ruedas ($u_R(t)$ la derecha y $u_L(t)$ la izquierda) [Lucas2001]. Los dos sensores lineales proporcionan la orientación (como haría una brújula) y los tres estados del sistema (como haría un sistema de balizamiento triular) mientras que el sistema no lineal devuelve la distancia al robot respecto al origen de coordenadas (siendo ésta por ejemplo la distancia medida por un sensor de ultrasonidos). Es importante destacar que para las funciones no lineales de este sistema, pequeños cambios de las variables de entrada no producirán cambios bruscos en las

variables de salida, hecho relevante en el recálculo de la información sensorial asociada a los sensores no lineales.

	Modelo		con $\mathbf{x}_t = (x(t) \ y(t) \ \theta(t))^T$ y
Sistema (NL)	I	$\begin{pmatrix} x(t+1) \\ y(t+1) \\ \theta(t+1) \end{pmatrix} = \begin{pmatrix} x(t) + \frac{u_R(t) + u_L(t)}{2} \cos\left(\theta(t) + \frac{u_R(t) - u_L(t)}{b}\right) \\ y(t) + \frac{u_R(t) + u_L(t)}{2} \sin\left(\theta(t) + \frac{u_R(t) - u_L(t)}{b}\right) \\ \theta(t) + \frac{u_R(t) - u_L(t)}{b} \end{pmatrix}$	$\mathbf{u}_t = \begin{pmatrix} u_R(t) \\ u_L(t) \end{pmatrix}, \mathbf{Q}_t = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.2\pi/180 \end{pmatrix}^2$
	R	$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, t) + \mathbf{v}_t$	
Sensor1 (L)	I	$z_{1\theta}(t+1) = \theta(t+1)$	$\mathbf{z}_{1,t+1} = z_{1\theta}(t+1), \mathbf{R}_{1,t+1} = (\pi/180)^2$
	R	$\mathbf{z}_{1,t+1} = h_1(\mathbf{x}_{t+1}, t+1) + \mathbf{v}_{1,t+1}$	
Sensor2 (L)	I	$\begin{aligned} z_{2x}(t+1) &= x(t+1) \\ z_{2y}(t+1) &= y(t+1) \\ z_{2\theta}(t+1) &= \theta(t+1) \end{aligned}$	$\mathbf{z}_{2,t+1} = \begin{pmatrix} z_{2x}(t+1) \\ z_{2y}(t+1) \\ z_{2\theta}(t+1) \end{pmatrix}, \mathbf{R}_{2,t+1} = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 2\pi/180 \end{pmatrix}^2$
	R	$\mathbf{z}_{2,t+1} = h_2(\mathbf{x}_{t+1}, t+1) + \mathbf{v}_{2,t+1}$	
Sensor3 (NL)	I	$z_{3d}(t+1) = x(t+1)^2 + y(t+1)^2$	$\mathbf{z}_{3,t+1} = z_{3d}(t+1), \mathbf{R}_{3,t+1} = 0.05^2$
	R	$\mathbf{z}_{3,t+1} = h_3(\mathbf{x}_{t+1}, t+1) + \mathbf{v}_{3,t+1}$	

Tabla C4-4. Ecuaciones del modelo del sistema y de los sensores para el primer caso no lineal. Tanto para el modelo del sistema como para los modelos de los sensores se presenta, en la primera columna, la ecuación explícita, para el modelo ideal (sin ruido), y las ecuaciones vectoriales reales, a las que se les ha añadido el término aditivo de ruido blanco. En la segunda columna se representan los vectores que relacionan el modelo ideal y real (salvo por el término aditivo de ruido) y la covarianza asociada al ruido para cada caso (\mathbf{Q}_t y $\mathbf{R}_{s,t}$)

A partir de las expresiones del modelo de transición y sensorial del sistema, de un estado inicial y del valor de la señal de control durante un intervalo de 80 segundos y con un periodo de muestreo de 0.1 segundos, se ha generado, mediante simulación, los datos experimentales básicos para todos los experimentos realizados durante esta sección. Estos datos están compuestos por una única trayectoria (x, y, θ) y las medidas realizadas para cada instante de muestreo por cada sensor sobre dicha trayectoria. La trayectoria generada durante el proceso de simulación ha sido almacenada para poder compararla con la trayectoria estimada por el algoritmo y es presentada, junto con la trayectoria ideal (generada por el modelo del sistema sin ruido) en las gráficas de la figura C4-16. La trayectoria (x, y, θ) generada a partir de la ecuación no lineal de transición y las ordenes de control elegidas para cada una de las ruedas es similar a la trayectoria utilizada como base experimental del análisis del funcionamiento de los algoritmos para sistemas con modelos lineales.

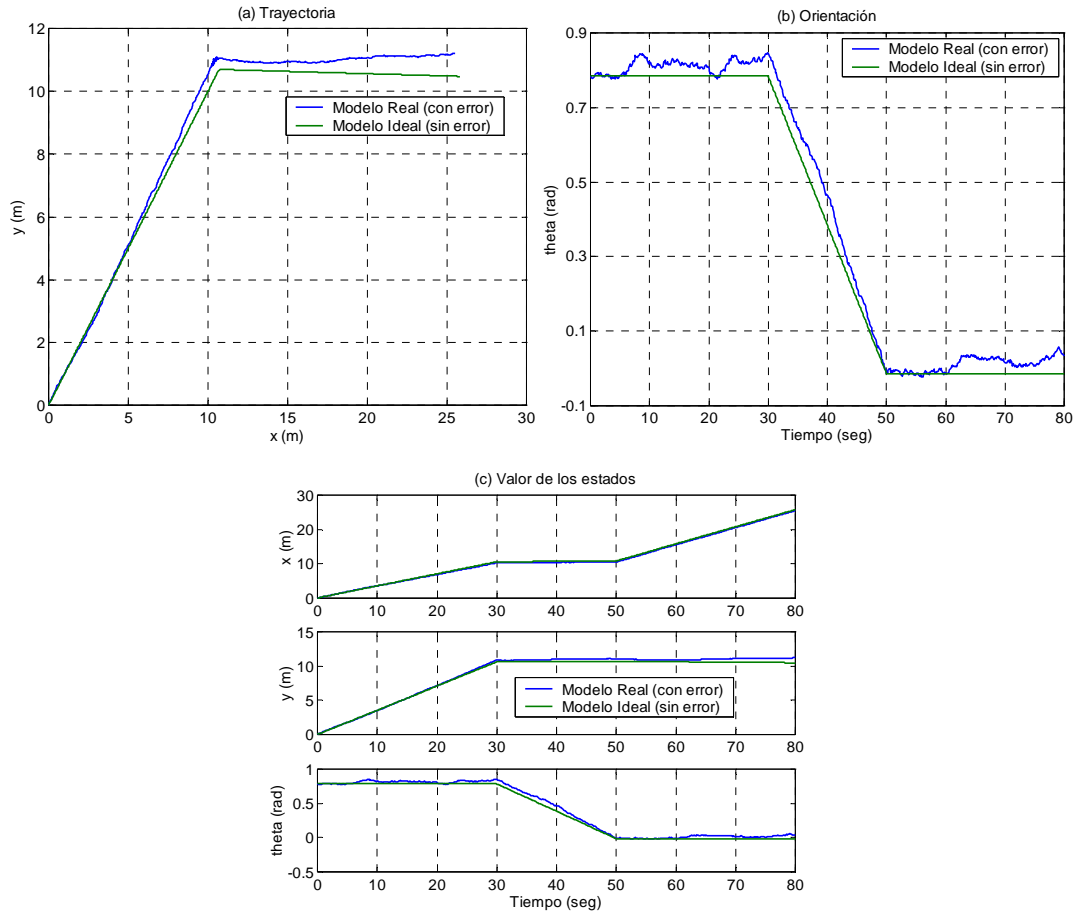


Figura C4-16. Trayectoria, orientación y valor de los estados utilizados durante el análisis del funcionamiento de IFAsynMixto para un sistema con funciones no lineales suaves.

Sobre estos datos experimentales se ha creado el grupo experimental IV, cuyas características se esquematizan en las gráficas de la figura C4-17. Este grupo está formado únicamente por dos experimentos: el (a) con medidas ordenadas y el (b) con las medidas del sensor2 retardadas 1 segundo durante todo el experimento. Sobre este grupo experimental se puede ejecutar el algoritmo IFAsynMixto con y sin etapa de validación, por lo que se pueden realizar 4 (2x2 posibilidades de validación) experimentos diferentes. No se han incluido experimentos donde hay sensores que no funcionan correctamente ni con medidas erróneas, como ocurre en el grupo experimental I, ya que las conclusiones extraídas en el análisis de funcionamiento del algoritmo lineal IFAsyn sobre la importancia de la etapa de validación cuando existen medidas erróneas o sobre la falta de medidas de un sensor son también aplicables al caso no lineal. En el estudio de esta sección se estudiará brevemente algunos de los aspectos analizados para el caso puramente lineal y se pondrán de manifiesto otros aspectos, asociados directamente con la falta de linealidad del modelo elegido y el tratamiento de este hecho por el algoritmo IFAsynMixto.

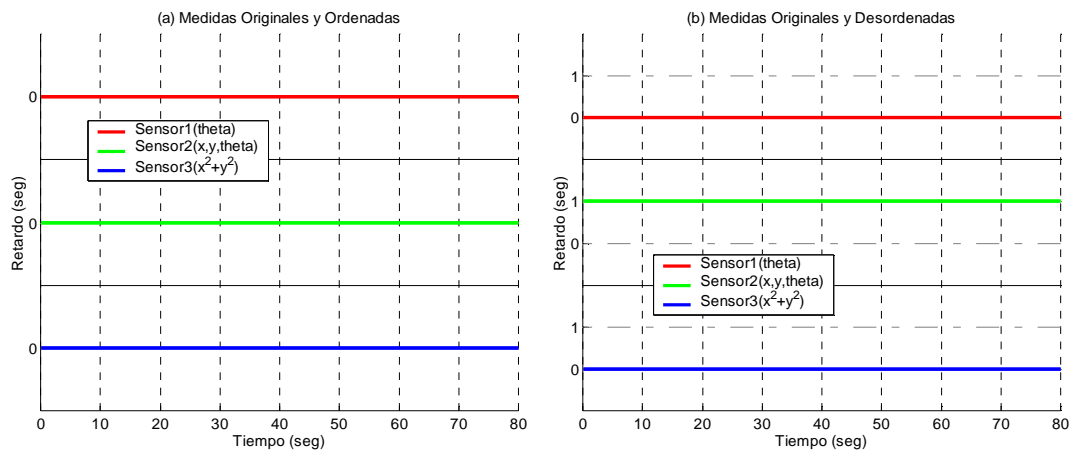


Figura C4-17. Caracterización de los experimentos del Grupo Experimental IV: Influencia de los retardos en los resultados obtenidos por IFAsynMixto.

Análisis de la influencia del desorden de las medidas en los resultados obtenidos y en su evolución según se asimilan las medidas desordenadas

En la primera sección de este capítulo, donde se estudian los resultados obtenidos por el algoritmo IFAsyn para sistemas con modelos lineales, se realizó un análisis exhaustivo de la influencia del desorden de la medida en los resultados obtenidos y de la evolución del valor del estado y su varianza para cada instante de muestreo t , según se van asimilando las medidas $\zeta_{s,k}$ realizadas en instantes anteriores ($k < t$) y que llegan al algoritmo temporalmente desordenadas. Las conclusiones de dicho análisis que se encuentran referidas al algoritmo IFAsyn pueden ser extrapoladas al algoritmo IFAsynMixto. Para comprobarlo, analizaremos alguno de los resultados obtenidos para los 4 (2x2 posibilidades de validación) experimentos del grupo experimental IV.

Para ver la influencia que el desorden de las medidas introduce en los resultados obtenidos, se compararán los resultados obtenidos por el algoritmo IFAsynMixto y el filtro de Kalman Extendido, que desecha las medidas temporalmente retardadas, para el caso ordenado (a) y desordenado (b), cuando se ejecutan los algoritmos sin (Sv) y con (Cv) etapa de validación. La comparación, que se presenta en la tabla C4-5, se hace de forma similar a la recogida en la tabla C4-3, es decir, presentando, para parejas de experimentos similares salvo por el orden de recepción de las medidas, la diferencia máxima existente entre los valores de la estima del estado (y de la varianza) y el número de medidas que han sufrido diferente suerte durante el proceso de validación.

Al igual que se observó en la comparativa del caso lineal, el filtro de Kalman Extendido presenta diferencias significativas para todos los experimentos ya que desecha todas las medidas del sensor2 (800), porque todas ellas llegan al algoritmo con un retardo de 1 segundo.

C4.2. Análisis del funcionamiento de los algoritmos para sistemas con modelos no lineales

Experimento	IFAsynMixto: Orden vs. Desorden			Kalman Extendido: Orden vs. Desorden		
	Estado	Var.	Valid	Estado	Var.	Valid
aSv vs. bSv	0	0	0	9.4e-2	2.5e-2	800
aCv vs. bCv	1.2e-2	6.2e-5	15	1.4e-1	7.9e-2	815

Tabla C4-5. Análisis de los resultados obtenidos por el mismo algoritmo cuando las medidas se encuentran ordenadas y desordenadas, para las 2 combinaciones posibles de los 4 ejemplos del Grupo Experimental IV.

En la primera columna se representan los experimentos, identificados con tres letras: la primera indica el experimento propuesto en la figura C4-17C a la que se refieren los resultados (*a* medidas originales ordenadas, *b* medidas originales desordenadas), y las dos siguientes si los experimentos se han realizado con etapa de validación (Cv) o no (Sv). Para cada algoritmo se representa la diferencia máxima existente entre el valor del estado estimado cuando las medidas se encuentran ordenadas y desordenadas, la diferencia máxima existente entre los valores de la varianza, y el número de medidas que han sido tratadas de forma diferente por cada algoritmo. En naranja se resaltan los casos cuya diferencia no puede considerarse despreciable.

Para el caso del algoritmo IFAsynMixto, el desorden no introduce ninguna diferencia cuando no se realiza la etapa de validación, mientras que introduce diferencias significativas entre los ejemplos con etapa de validación. En las gráficas de la figura C4-18, se puede observar que en el ejemplo con validación las diferencias significativas entre los valores de los estados (gráfica (b)) y la varianza (gráfica (c)) se producen en torno a los instantes de muestreo en los que el orden de recepción de las medidas ha influido en la etapa de validación (gráfica (a)). Estas diferencias se deben a que una vez que una medida es considerada válida o errónea, las medidas realizadas en instantes anteriores y recibidas por el algoritmo en instantes posteriores no modificarán la decisión ya tomada.

Para el ejemplo propuesto se puede observar como los cambios en la validación de cada sensor producen cambios significativos en la varianza asociada a los estados sobre los que realizan medidas: el sensor2, que mide el valor de los tres estados, sufre un único cambio de validación que se ve reflejado en la diferencia de varianza de los tres estados, mientras que el sensor1, que mide la orientación del robot, sufre 14 cambios de validación que se encuentran reflejados en la diferencia de varianza de del tercer estado.

El reflejo de los cambios en la validación en la diferencia del valor estimado de los estados sufre un comportamiento diferente, ya que el estado de la orientación se utiliza, en las etapas de predicción, para estimar la posición (estados 1 y 2) del móvil. Por lo tanto, el cambio en la validación de las medidas del sensor1 (que mide únicamente la orientación) se encuentra reflejado en las diferencias de los valores de los tres estados.

Este fenómeno no se observó durante los experimentos analizados en el ejemplo puramente lineal, cuyas ecuaciones se encuentran recogidas en la tabla C4-1, debido a que en el modelo de transición elegido para ese caso el valor de ningún estado dependía de los valores de los otros estados. Es decir, el comportamiento observado en este ejemplo no es debido a la falta de linealidad del modelo de transición, sino a la interdependencia existente entre los estados del sistema. En experimentos sobre sistemas puramente lineales con un modelo de transición donde el valor de un

estado dependa de los valores de otros estados se observarán comportamientos análogos a los observados en el sistema no lineal.

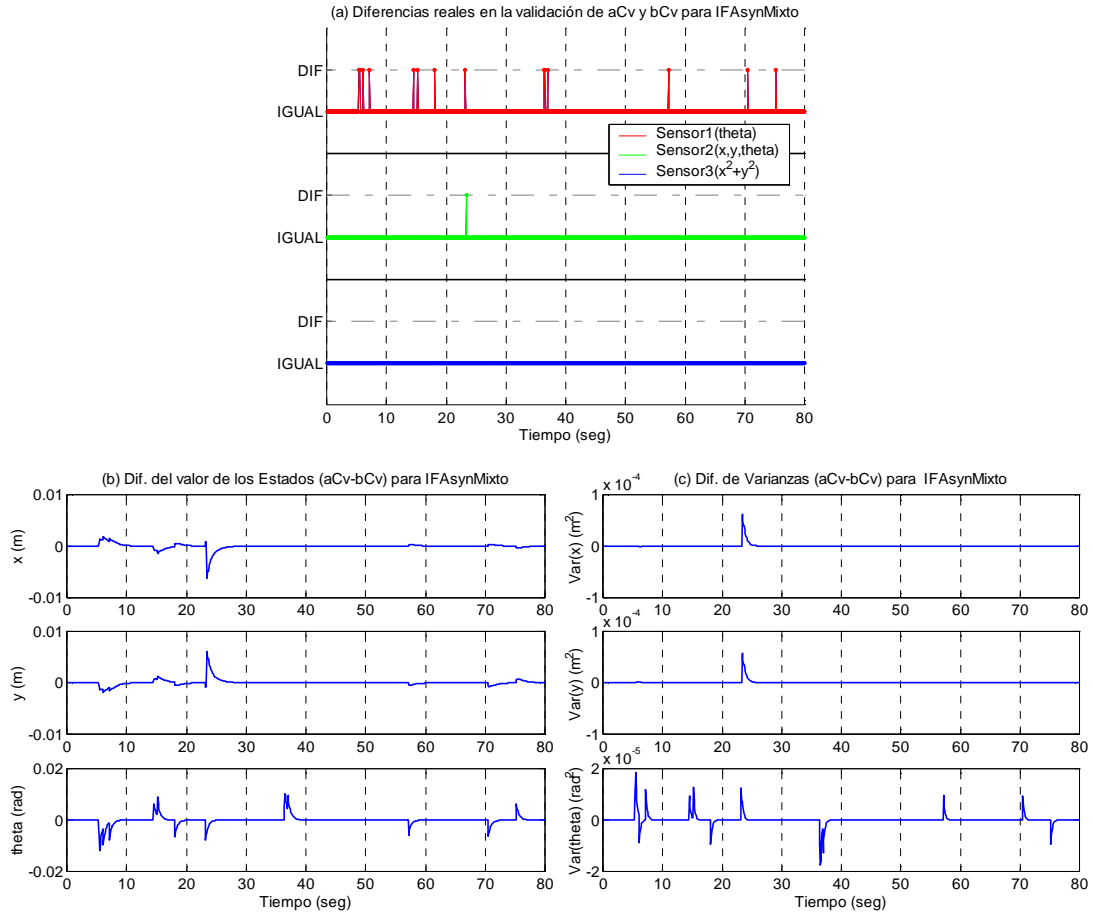


Figura C4-18. Discrepancia existente entre los ejemplos del grupo experimental IV aCv (medidas originales ordenadas y algoritmo con etapa de validación) y bCv (medidas originales desordenadas y algoritmo con etapa de validación) para el algoritmo IFAsynMixto. En la gráfica (a) se representa la diferente suerte que han sufrido durante la validación las medidas de los tres sensores, y en las gráficas (b) y (c) las discrepancias entre el valor de los estados y varianza respectivamente.

Otro aspecto importante analizado para el caso puramente lineal es la evolución del valor del estado y de la varianza para cada instante de muestreo según se va asimilando la información medida en instantes anteriores y que llega al algoritmo temporalmente retardada. En las gráficas de la figura C4-19 se representa dicha evolución, para el experimento b del grupo experimental IV cuando se ejecuta el algoritmo IFAsynMixto sin y con etapa de validación. Para cada uno de los valores representados se presentan tres curvas, las cuales representan para cada instante de muestreo t , tres momentos diferentes del proceso de asimilación: la curva *Inicio* contiene los valores obtenidos con la información sensorial que ha sido medida y recibida por el algoritmo hasta el instante t , la curva *Final* recoge los valores obtenidos una vez que se ha asimilado toda la información retardada medida hasta t , y la curva *Medio* representa los valores calculados tras la

asimilación de la mitad de la información medida hasta el instante t que se recibe en instantes posteriores a t .

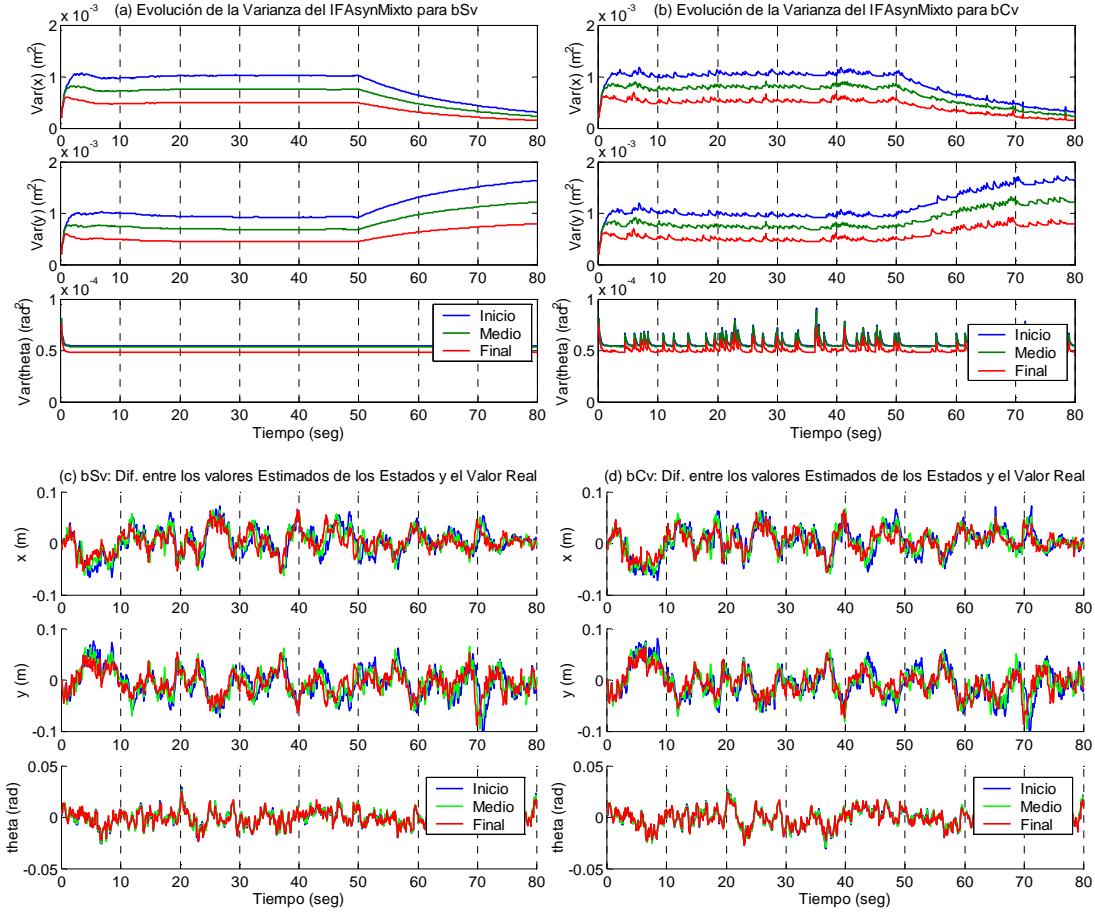


Figura C4-19. Evolución del valor de la varianza (gráficas (a) y (b)) y de la diferencia entre el valor estimado del estado y el real (gráficas (c) y (d)), para el experimento b del grupo experimental IV, obtenida por el algoritmo IFAsynMixto sin y con validación, según los algoritmos van asimilando la información retardada. La curva Inicio corresponde a los valores que el algoritmo calcula con la información sensorial disponible hasta el correspondiente instante de muestreo. La curva Final representa los valores obtenidos una vez que se ha asimilado toda la información retardada. En la curva Medio se observan los valores estimados con la mitad de la información retardada respecto a cada instante de muestreo.

En las gráficas (a) y (b) se representa la evolución de la varianza obtenida por el algoritmo IFAsynMixto sin y con etapa de validación respectivamente. Al igual que en los ejemplos lineales se observa que según aumenta la información sensorial asimilada, que es menor en el caso *Inicial* que en el caso *Medio*, y en el caso *Medio* que en el *Final*, la varianza disminuye. Los picos que aparecen en la gráfica (b) son debidos a las medidas rechazadas por la etapa de validación del algoritmo.

En las gráficas (c) y (d) se representa la diferencia entre el valor del estado estimado por el algoritmo y su valor real, sin y con etapa de validación respectivamente. Como ya se comentó en la sección del análisis del sistema lineal, la utilización de nueva información sensorial puede

modificar el valor del estado en un sentido o en el contrario. Aun así, según aumenta la información asimilada, en promedio, la discrepancia entre el valor estimado del estado y el valor real disminuye, hecho que se puede observar en las gráficas (c) y (d), donde los valores de la curva *Final* (en rojo) están normalmente más próximos al cero que los de las curvas *Medio* (en verde) e *Inicio* (en azul).

Con estos experimentos y estudio se ha comprobado que los resultados del algoritmo IFAsynMixto para el ejemplo propuesto en la tabla C4-4 y los del algoritmo IFAsyn para el sistema propuesto en la tabla C4-1 presentan comportamientos análogos cuando se analiza la influencia del desorden de las medidas en los resultados obtenidos, en el proceso de validación, y en la evolución de los resultados según se va asimilando la información que llega al algoritmo de forma temporalmente retardada.

A continuación se analizan algunas características propias del algoritmo IFAsynMixto cuando éste es utilizado para estimar el estado y la covarianza de un sistema con sensores no lineales.

Necesidad del término corrector de la medida para la asimilación sensorial proveniente de los sensores no lineales

Las expresiones de asimilación del filtro de Información para los sensores no lineales, que se encuentran recogidas en el apéndice I, utilizan, en vez de la medida original $\zeta_{s,t+1}$ de las expresiones de asimilación lineales, una medida corregida $\zeta_{s,t+1}^C$ con la discrepancia existente entre el valor de la medida estimado por la ecuación del sensor $h_s(\hat{\mathbf{x}}_{t+1|AASIM \leq t}, t+1)$ y el valor estimado por el modelo linealizado $\mathbf{H}_{s,t+1}\hat{\mathbf{x}}_{t+1|AASIM \leq t}$.

En esta sección se comprueba la necesidad del término corrector de la medida para el buen funcionamiento del filtro de Información Extendido. El algoritmo IFAsynMixto utiliza la variable *SLineal* para decidir para qué sensores hay que incluir el término corrector. Este mecanismo nos permitirá analizar fácilmente la influencia del término corrector. Para cada experimento ejecutaremos el algoritmo dos veces: la variable *SLineal* de la primera ejecución indicará que se está trabajando con dos sensores lineales (sensor1 y sensor2) y uno no lineal (sensor3), mientras que en la segunda ejecución le haremos creer al IFAsynMixto que está trabajando con tres sensores lineales, a pesar de que para ambos casos, las ecuaciones del modelo de los sensores son las recogidas en la tabla C4-4. De esta forma, el algoritmo IFAsynMixto estimará el estado del sistema a partir de la misma información sensorial incluyendo el término corrector cuando considera que el sensor3 es no lineal y excluyéndolo cuando considera que el sensor3 es lineal, a pesar de no serlo realmente.

El análisis de este apartado se realizará para los experimentos (a) y (b) del grupo experimental IV, ejecutando el algoritmo IFAsynMixto sin etapa de validación. De esta forma, evitaremos que la

etapa de validación empiece a rechazar medidas diferentes cuando el algoritmo IFAsynMixto se ejecuta para el mismo ejemplo con término corrector y sin él.

En las gráficas de la figura C4-20 se presentan los resultados obtenidos para los dos experimentos y las dos ejecuciones del algoritmo. Debido a la falta de etapa de validación, los resultados obtenidos para cada experimento cuando se ejecuta el filtro sintonizado de la misma forma (con o sin término corrector) son iguales porque este algoritmo sin etapa de validación obtiene los mismos resultados cuando las medidas se encuentran ordenadas (experimento a) y desordenadas (experimento b).

En la gráfica (a), donde se representa la diferencia entre el valor estimado y el real, se puede observar como la falta del término corrector provoca un aumento continuo de la discrepancia entre el valor real y el estimado de los estados x e y . En la gráfica (b) se observa bajo el formato de trayectoria los valores estimados para las posiciones x e y cuando el filtro IFAsynMixto utiliza el término corrector (y por lo tanto el algoritmo trata la información sensorial del sensor3 no lineal como lo que es, información procedente de un sensor no lineal) y los valores estimados cuando no se utiliza el término corrector durante la asimilación (es decir, cuando el algoritmo considera que está trabajando con un sensor lineal a pesar de que éste no lo sea). De esta forma, resulta más sencillo observar la importancia de la discrepancia existente entre los resultados obtenidos por el filtro sintonizado correcta e incorrectamente.

La discrepancia entre el valor real y el estimado del estado θ es similar cuando el filtro obtiene el valor estimado con y sin término corrector debido a que el sensor no lineal (sensor3) únicamente proporciona información sensorial referente a los estados x e y .

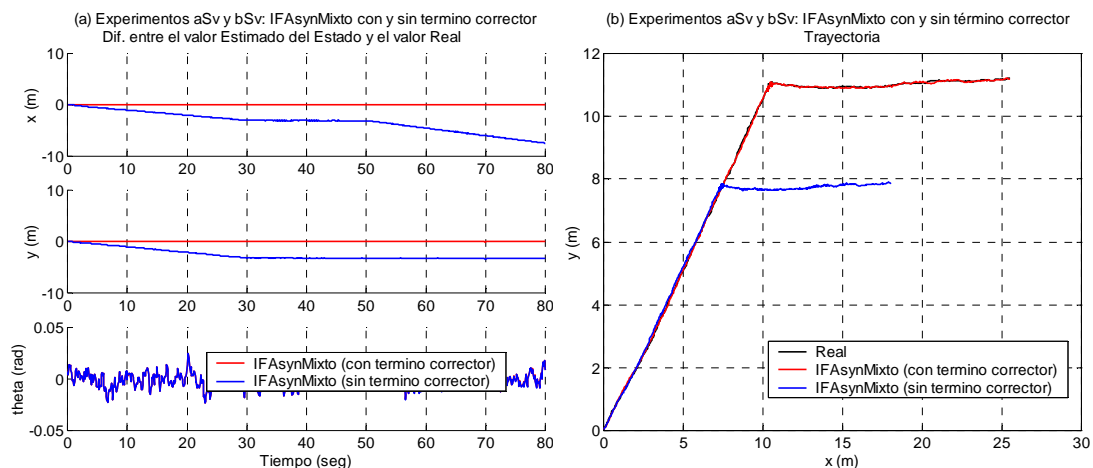


Figura C4-20. Trayectoria y diferencia entre el valor real del estado y el estimado del sistema, para los experimentos a y b del grupo experimental IV, cuando se ejecuta el algoritmo IFAsynMixto sintonizado de tal forma que en un caso considera que el sensor3 es no lineal (para que utilice el término corrector de la medida en la asimilación), y en el otro considera que el sensor3 es lineal (a pesar de no serlo, y no utiliza el término corrector). En ambos casos, el algoritmo se ejecuta sin etapa de validación. En la gráfica (a) se representa la diferencia entre el valor estimado del estado (con y sin término corrector) y el valor real del estado. En la gráfica (b) se representan las trayectorias reales, y estimadas con y sin término corrector.

Por lo tanto, se ha comprobado la necesidad del término corrector de la medida para la asimilación, tanto ordenada (ejemplo a) como desordenada (ejemplo b), de la información procedente de un sensor no lineal.

Necesidad del recálculo de la información sensorial asociada a los sensores no lineales cuando llega al algoritmo información retardada proveniente de cualquier sensor

La dependencia de la información sensorial de los sensores no lineales del valor estimado del estado $\hat{\mathbf{x}}_{t|AASIM \leq t-1}$ (a través del jacobiano y del término corrector de medida) es la causante de la necesidad del recálculo de la información sensorial ($\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$) asociada a estos sensores cuando el algoritmo IFAsynMixto asimila las medidas retardadas tomadas por cualquier sensor en instantes anteriores a t . Esto es debido a que la asimilación de las medidas realizadas antes del instante t y que llegan al algoritmo después de dicho instante modificará el valor de $\hat{\mathbf{x}}_{t|AASIM \leq t-1}$, y el nuevo valor de esta variable modificará a su vez el valor de $\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$ correspondiente a las medidas de los sensores no lineales ya asimiladas por el algoritmo.

Para analizar la influencia del recálculo de la información sensorial asociada a los sensores no lineales en los valores estimados para el estado y su varianza, utilizaremos el algoritmo IFAsynMixto y una versión modificada del mismo, identificada como IFAsynMixtoEsp y recogida en el apéndice II, sobre el ejemplo con medidas desordenadas del grupo experimental IV. En ninguna de la ejecución de las dos versiones del algoritmo C3-3 (original y modificado) se utilizará la etapa de validación, para evitar observar en el experimento la influencia que ésta tiene sobre la asimilación de medidas desordenadas. Es decir, las dos versiones del algoritmo C3-3 se ejecutarán sin validación sobre el ejemplo con medidas desordenadas (bSv). En dicho ejemplo la medida que llega temporalmente retardada es la proporcionada por el sensor2, hecho que provoca la necesidad del recálculo de la información sensorial ($\mathbf{i}_{s,t}$ e $\mathbf{I}_{s,t}$) asociada al único sensor no lineal, el sensor3, y de la reasimilación de dicha información sensorial.

En la versión modificada del algoritmo (IFAsynMixtoEsp) la información asociada a los sensores no lineales (el sensor3 en este caso) únicamente se calcula cuando la medida es recibida por el algoritmo, utilizando para dicho cálculo la expresión que incluye el término corrector asociado a los sensores no lineales. Una vez calculada, la asimilación de medidas retardadas no modificará el valor de la información asociada a la medida no lineal, hecho que reducirá considerablemente el cálculo de la versión modificada del algoritmo C3-3 frente a su versión original, que tiene que recalcular y reasimilar la información sensorial asociada a las medidas no lineales cuando el algoritmo recibe medidas retardadas asociadas a instantes de muestreo anteriores. En el apéndice II se recogen las dos versiones del algoritmo C3-3, dispuestas en sendas columnas de la tabla A2-2, con el objeto de poder detectar fácilmente las diferencias existentes entre el algoritmo IFAsynMixto e IFAsynMixtoEsp.

Ambas versiones del algoritmo serán sintonizadas con el valor correcto, para este ejemplo, de la variable *SLineal*. Es decir, en ella se indicará que el sensor1 y el sensor2 son lineales mientras que el sensor3 no lo es.

En las gráficas de la figura C4-21 se recogen los resultados obtenidos sobre el ejemplo b del grupo experimental IV por cada una de las dos versiones del algoritmo C3-3: la original (IFAsynMixto) y la modificada (IFAsynMixtoEsp).

En la gráfica (a) se representan los valores de la variable de estado estimados por cada versión del algoritmo, y a simple vista no se puede apreciar ninguna discrepancia ya que las curvas asociadas a cada estado y versión del algoritmo coinciden. Con el objeto de poder observar la discrepancia existente, en la gráfica (c) se representa la diferencia entre los valores estimados por ambos algoritmos. En dicha gráfica se puede observar que la discrepancia máxima, en valor absoluto, entre el valor estimado para los estados x e y es menor que $2e-3m$ (2 mm) mientras que el valor absoluto de la discrepancia existente entre los valores estimados para θ es menor que $2e-5$ radianes.

Para medir la importancia real de la discrepancia utilizaremos el valor de la desviación estándar, cuyos valores se pueden calcular a partir de la varianza representada en la gráfica (b) de la misma figura. La desviación estándar máxima de los estados x e y es menor a los 3 cm y para θ inferior a 0.01 radianes.

La medida de la discrepancia relativa a cada estado e instante de muestreo la obtendremos dividiendo el valor absoluto de la discrepancia entre los valores del estado estimados por las dos versiones del algoritmo entre el máximo de la desviación estándar calculada por las dos versiones del algoritmo para el mismo instante. El resultado de esta operación se encuentra recogido en la gráfica (e):

- Para los estados x e y la discrepancia entre los estados es inferior a la décima parte de su desviación estándar. Por lo tanto, podría considerarse despreciable, ya que la corrección debido al recálculo de la información asociada a los sensores no lineales se encuentra englobada por debajo de 10 veces el valor de la desviación.
- Para el estado que almacena la orientación (θ) la razón entre la discrepancia entre los estados y su desviación estándar es inferior a $1.5e-3$. Es decir, la corrección en la orientación debido al cálculo de la información asociada a los sensores no lineales es significativamente menor que el valor de la desviación asociada a dicho estado.

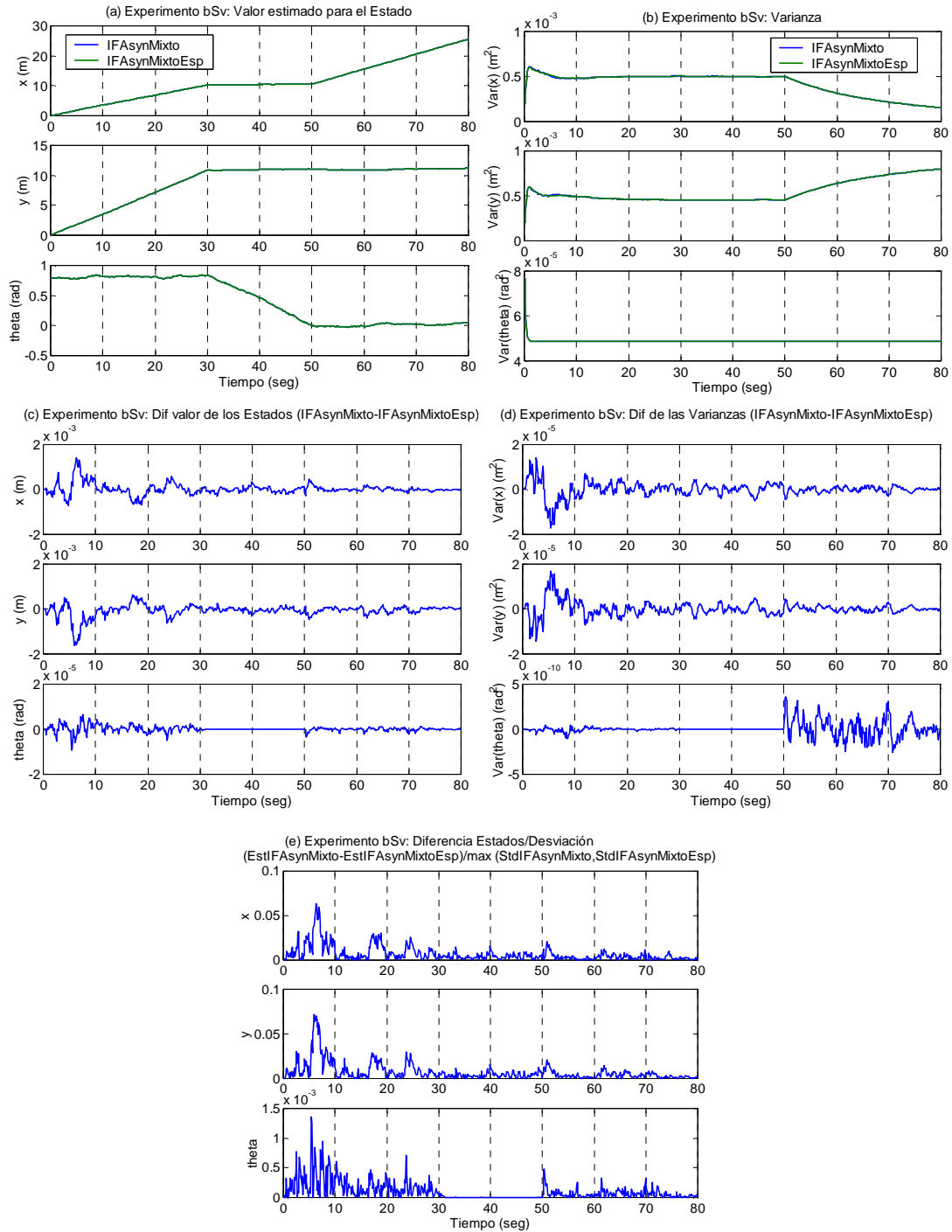


Figura C4-21. Resultados del algoritmo C3-3 (IFAsynMixto) y de una versión modificada del mismo (IFAsynMixtoEsp, que no recalcula la información sensorial asociada a los sensores no lineales cuando llega información sensorial retardada), cuando ambos algoritmos son ejecutados sin etapa de validación sobre el experimento b del grupo experimental IV. En la gráfica (a) se representan los valores del estado estimados por los dos algoritmos y en la gráfica (b) las varianzas obtenidas. En la gráfica (c) se representa la discrepancia entre los valores del estado estimados por los dos algoritmos y en la gráfica (d) la discrepancia de las varianzas. En la gráfica (e) se representa la importancia de la discrepancia entre los valores estimados del estado por cada versión del algoritmo relativa al valor de la desviación estándar máxima calculada.

La discrepancia entre los valores de la varianza obtenida para los estados x e y por las dos versiones del algoritmo C3-3 puede verse a simple vista en la gráfica (b) de la figura C4-21. Aun así, ha sido

representada en la gráfica (d) de la misma figura, como la diferencia entre la varianza obtenida por las dos versiones del algoritmo. En ella se puede observar que la discrepancia máxima entre la varianza de los estados x e y es menor, en valor absoluto, que $2e-6$, lo que representa la centésima parte de los valores obtenidos para la varianza de dichos estados. Para el caso de la orientación, la discrepancia máxima entre la varianza es menor que $5e-10$, lo que representa diezmilésima parte de los valores de la varianza obtenidos para la orientación. Por lo tanto, la discrepancia en la varianza puede considerarse despreciable para los tres estados, y sobre todo, para la orientación.

La razón de la discrepancias en los valores estimados por las dos versiones del algoritmo para los estados x e y se encuentra en el recálculo de la información sensorial y reasimilación del sensor no lineal, que únicamente proporciona información sobre dichos estados, para IFAsynMixto y la falta de dicho recálculo en IFAsynMixtoEsp. Las discrepancias en los estados x e y se propagan, en menor grado, al estado θ debido a la dependencia que existe entre los diferentes estados en la ecuación de transición de estados.

De los resultados observados para este ejemplo, se puede concluir que la utilización de la versión modificada del algoritmo C3-3 en vez de la original es ventajosa en este caso ya que se reduce el número de operaciones a costa de introducir un error despreciable en los valores estimados del estado y la varianza. Esta afirmación se sustenta, además de en los resultados observados durante numerosas simulaciones del sistema, en las características de las ecuaciones que modelan el sistema elegido. La función no lineal de medida del sensor3 presenta un comportamiento suave, que ante pequeños cambios de la variable de estado \mathbf{x} debidos a la asimilación retardada de las medidas del sensor2, introduce modificaciones despreciables en el término corrector de la medida y en el jacobiano de la función medida, y por lo tanto, cambios insignificantes en la información de la medida del sensor no lineal.

En las ecuaciones de la siguiente tabla se analiza, matemáticamente, la influencia del cambio de valor de la variable del estado \mathbf{x} , debido a la asimilación de medidas temporalmente retardadas, en la información sensorial ($\mathbf{i}_{s,t+1}$ e $\mathbf{I}_{s,t+1}$) proporcionada por cualquier sensor no lineal. En las dos primeras filas se recoge el cálculo de la información sensorial asociada al sensor, a partir de la misma medida $\zeta_{s,t+1}$ y de dos valores diferentes del estado: el primero $\hat{\mathbf{x}}_{t+1|t}$ representa el valor estimado por el algoritmo en el instante que recibió la medida y la incorporó como evidencia, y el segundo $\hat{\mathbf{x}}_{t+1|t} + \Delta\mathbf{x}$ recoge el valor del estado cuando se recibió la medida junto al incremento (o decremento) debido a la asimilación retardada de medidas previas a $t+1$. En la fila tercera se calcula la corrección que la reasimilación de la medida introduce en el valor de la información sensorial. Se puede observar como esta corrección, que es la responsable, una vez asimilada la nueva medida, de las discrepancias existentes entre los resultados obtenidos por los algoritmos IFAsynMixto e IFAsynMixtoEsp, depende de la influencia que el $\Delta\mathbf{x}$ tiene en la función de medida y en el jacobiano de dicha función. El resultado de la última fila se obtiene suponiendo que la

influencia del $\Delta \mathbf{x}$ en el jacobiano es despreciable: en ese caso la varianza de la información sensorial $\mathbf{I}_{s,t+I}$ no se verá modificada mientras que la información sensorial pasará a depender de la influencia de $\Delta \mathbf{x}$ en la ecuación de medida y del tamaño de $\Delta \mathbf{x}$.

ETAPA 1: Cálculo de la información sensorial asociada a un sensor no lineal, a partir de la medida ($\zeta_{s,t+1}$) y del valor del estado ($\hat{\mathbf{x}}_{t+I/t}$)		$\mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}} = (\mathbf{J}h_s)_x(\hat{\mathbf{x}}_{t+I/t}, t+I)$ $\zeta_{s,\hat{\mathbf{x}}_{t+I/t}}^C = \zeta_{s,t+I} - h_s(\hat{\mathbf{x}}_{t+I/t}, t+I) + \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}} \hat{\mathbf{x}}_{t+I/t}$ $\mathbf{i}_{s,\hat{\mathbf{x}}_{t+I/t}} = \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}^T \mathbf{R}_{s,t+I}^{-1} \zeta_{s,\hat{\mathbf{x}}_{t+I/t}}^C$ $\mathbf{I}_{s,\hat{\mathbf{x}}_{t+I/t}} = \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}^T \mathbf{R}_{s,t+I}^{-1} \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}$
ETAPA 2: Recálculo de la información sensorial asociada al mismo sensor, a partir del mismo valor de medida ($\zeta_{s,t+1}$) y del nuevo valor del estado ($\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}$), que se encuentra modificado un $\Delta \mathbf{x}$ debido a la asimilación retardada de medidas previas a t+1.		$\mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}} = (\mathbf{J}h_s)_x(\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}, t+I)$ $\zeta_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^C = \zeta_{s,t+I} - h_s(\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}, t+I) + \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}(\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x})$ $\mathbf{i}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}} = \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^T \mathbf{R}_{s,t+I}^{-1} \zeta_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^C$ $\mathbf{I}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}} = \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^T \mathbf{R}_{s,t+I}^{-1} \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}$
Discrepancia entre la información sensorial asimilada en las ETAPAS 1 y 2.	Real	$\Delta \mathbf{i}_{s,t+I} = \mathbf{i}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}} - \mathbf{i}_{s,\hat{\mathbf{x}}_{t+I/t}}$ $= \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^T \mathbf{R}_{s,t+I}^{-1} \zeta_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^C - \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}^T \mathbf{R}_{s,t+I}^{-1} \zeta_{s,\hat{\mathbf{x}}_{t+I/t}}^C$ $\Delta \mathbf{I}_{s,t+I} = \mathbf{I}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}} - \mathbf{I}_{s,\hat{\mathbf{x}}_{t+I/t}}$ $= \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^T \mathbf{R}_{s,t+I}^{-1} \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}} - \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}^T \mathbf{R}_{s,t+I}^{-1} \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}$
	Suponiendo $\mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}^T \cong \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^T$	$\Delta \mathbf{i}_{s,t+I} \cong \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}^T \mathbf{R}_{s,t+I}^{-1} (\zeta_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^C - \zeta_{s,\hat{\mathbf{x}}_{t+I/t}}^C) =$ $= \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}^T \mathbf{R}_{s,t+I}^{-1} (h_s(\hat{\mathbf{x}}_{t+I/t}, t+I) - h_s(\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}, t+I) + \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}} \Delta \mathbf{x})$ $\Delta \mathbf{I}_{s,t+I} = \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}}^T \mathbf{R}_{s,t+I}^{-1} \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t} + \Delta \mathbf{x}} - \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}}^T \mathbf{R}_{s,t+I}^{-1} \mathbf{H}_{s,\hat{\mathbf{x}}_{t+I/t}} \cong 0$

Tabla C4-6. Cálculo de la discrepancia introducida en la reasimilación de la información sensorial asociada a los sensores no lineales cuando existe información temporalmente retardada.

En el sistema ejemplo recogido en la tabla C4-4, las funciones del modelo del sistema y de los sensores son lo suficientemente suaves para que los pequeños cambios del estado $\Delta \mathbf{x}$ causados por la incorporación de la información retardada proveniente del sensor2 tengan una influencia despreciable en la función de medida del sensor3 y su jacobiano correspondiente. Como consecuencia de estas características, la influencia del recálculo de la información sensorial del sensor no lineal 3 es despreciable.

▪ Análisis de los resultados para un sistema con funciones no lineales abruptas

Por último, para ver la importancia del recálculo de la información sensorial asociada a los sensores no lineales en un sistema diferente al planteado en la tabla C4-4, se ha buscado un sistema alternativo, cuyos funciones de medida sean lo suficientemente abruptas para que las diferencias obtenidas entre el algoritmo C3-3 original (IFAsynMixto) y modificado (IFAsynMixtoEsp) sean

significativas. Las ecuaciones de este nuevo sistema se recogen en la siguiente tabla. El modelo del sistema y del sensor1 (brújula) son los mismos que para el ejemplo no lineal con funciones no lineales suaves, mientras que las ecuaciones del sensor2 y del sensor3 han sido modificadas. El sensor2, que proporcionará las medidas retardadas, ha dejado de ser puramente lineal para convertirse en un sensor no lineal que mide directamente el estado θ del sistema y el seno de los estados x e y . El sensor3 sigue siendo no lineal, pero en vez de medir la distancia euclídea respecto al origen de coordenadas, se encuentra modelado por una función no lineal, en la que se suma el valor cúbico de los senos de los estados x e y . Además de los cambios asociados a las ecuaciones de medida, la covarianza asociada a los mismos ha sido modificada, reduciendo su valor, para que el algoritmo C3-3 considere que la información sensorial proporcionada por los sensores es más exacta, y por lo tanto, tenga un peso mayor en el valor estimado del estado.

Es importante subrayar que las ecuaciones elegidas para estos dos nuevos sensores no tienen como objeto modelar el funcionamiento de ningún sensor real, sino solamente poner de manifiesto la necesidad del recálculo de la información sensorial asociada a los sensores no lineales cuando al algoritmo IFAsynMixto llegan medidas temporalmente retardadas.

		Modelo	con $\mathbf{x}_t = (x(t) \ y(t) \ \theta(t))^T$ y
Sistema (NL)	I	$\begin{pmatrix} x(t+1) \\ y(t+1) \\ \theta(t+1) \end{pmatrix} = \begin{pmatrix} x(t) + \frac{u_R(t) + u_L(t)}{2} \cos\left(\theta(t) + \frac{u_R(t) - u_L(t)}{b}\right) \\ y(t) + \frac{u_R(t) + u_L(t)}{2} \sin\left(\theta(t) + \frac{u_R(t) - u_L(t)}{b}\right) \\ \theta(t) + \frac{u_R(t) - u_L(t)}{b} \end{pmatrix}$	$\mathbf{u}_t = \begin{pmatrix} u_R(t) \\ u_L(t) \end{pmatrix}, \mathbf{Q}_t = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.2\pi/180 \end{pmatrix}^2$
	R	$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, t) + \mathbf{v}_t$	
Sensor1 (L)	I	$z_{I\theta}(t+1) = \theta(t+1)$	$\mathbf{z}_{I,t+1} = z_{I\theta}(t+1), \mathbf{R}_{I,t+1} = (\pi/180)^2$
	R	$\mathbf{z}_{I,t+1} = h_I(\mathbf{x}_{t+1}, t+1) + \mathbf{v}_{I,t+1}$	
Sensor2 (NL)	I	$\begin{aligned} z_{2x}(t+1) &= \sin(x(t+1)) \\ z_{2y}(t+1) &= \sin(y(t+1)) \\ z_{2\theta}(t+1) &= \theta(t+1) \end{aligned}$	$\mathbf{z}_{2,t+1} = \begin{pmatrix} z_{2x}(t+1) \\ z_{2y}(t+1) \\ z_{2\theta}(t+1) \end{pmatrix}, \mathbf{R}_{2,t+1} = \begin{pmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 2\pi/180 \end{pmatrix}^2$
	R	$\mathbf{z}_{2,t+1} = h_2(\mathbf{x}_{t+1}, t+1) + \mathbf{v}_{2,t+1}$	
Sensor3 (NL)	I	$z_{3d}(t+1) = \sin^3(2x(t+1)) + \sin^3(2y(t+1))$	$\mathbf{z}_{3,t+1} = z_{3d}(t+1), \mathbf{R}_{2,t+1} = 0.01^2$
	R	$\mathbf{z}_{3,t+1} = h_3(\mathbf{x}_{t+1}, t+1) + \mathbf{v}_{3,t+1}$	

Tabla C4-7. Ecuaciones del modelo del sistema y de los sensores para el segundo caso no lineal. Tanto para el modelo del sistema como para los modelos de los sensores se presenta, en la primera columna, la ecuación explícita, para el modelo ideal (sin ruido) y las ecuaciones vectoriales reales, a las que se les ha añadido el término aditivo de ruido blanco. En la segunda columna se representan los vectores que relacionan el modelo ideal y real (salvo por el término aditivo de ruido) y la covarianza asociada al ruido para cada caso (\mathbf{Q}_t y $\mathbf{R}_{s,t}$).

Al igual que en los ejemplos anteriores, a partir de las expresiones del modelo de transición y sensorial del sistema, de un estado inicial y del valor de la señal de control durante un intervalo de 80 segundos y con un periodo de muestreo de 0.1 segundos, se ha generado, mediante simulación, los datos experimentales básicos para todos los experimentos realizados durante esta sección. Estos datos están compuestos por una única trayectoria (x, y, θ) y las medidas realizadas para cada instante de muestreo por cada sensor sobre dicha trayectoria. La trayectoria generada durante el proceso de simulación ha sido almacenada para poder compararla con las trayectorias estimadas por los algoritmos y es presentada, junto con la trayectoria ideal (generada por el modelo del sistema sin ruido) en las gráficas de la figura C4-22. Las señales de control elegidas y el estado inicial son los mismos que los del ejemplo con funciones no lineales suaves, por lo que la trayectoria ideal coincide.

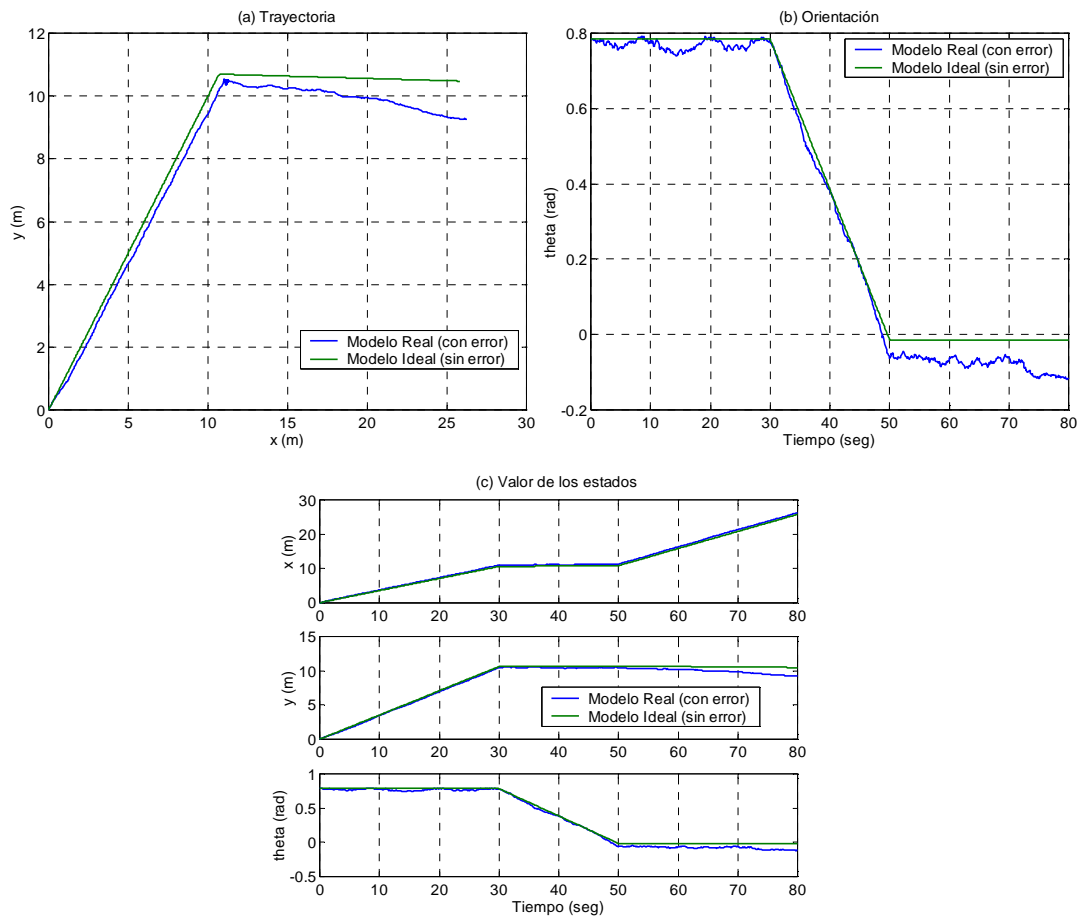


Figura C4-22. Trayectoria, orientación y valor de los estados utilizados durante el análisis del funcionamiento del algoritmo C3-3 para un sistema con funciones no lineales abruptas.

Sobre estos datos experimentales se ha creado el grupo experimental V. Este grupo está formado únicamente por dos experimentos: el identificado como (a) con medidas ordenadas y el etiquetado con (b) con las medidas del sensor2 retardadas 1 segundo durante todo el experimento. Sobre este grupo experimental se pueden ejecutar los algoritmos IFAsynMixto e IFAsynMixtoEsp con y sin

etapa de validación, por lo que se pueden realizar 4 (2x2 posibilidades de validación) experimentos diferentes para cada algoritmo. En el análisis de esta sección únicamente se recogen los resultados de la ejecución de los algoritmos IFAsynMixto e IFAsynMixtoEsp sin etapa de validación, debido a que las conclusiones extraídas para el resto del estudio realizado sobre sistemas con funciones no lineales suaves pueden ser aplicadas a este ejemplo. En la gráfica de la figura C4-23 se representan las características del experimento (b) del grupo experimental V.

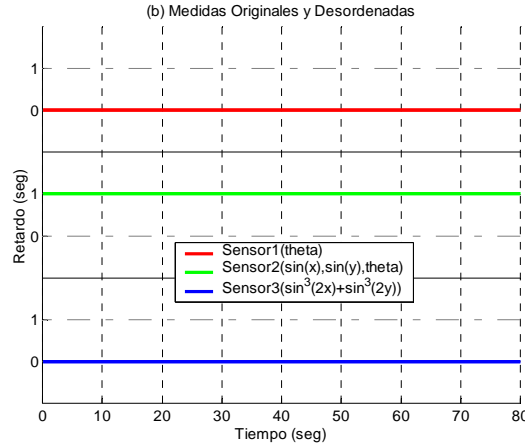


Figura C4-23. Caracterización de los experimentos (b) del Grupo Experimental V: Influencia de los retardos en los resultados obtenidos por el algoritmo C3-3 original (IFAsynMixto) y modificado (IFAsynMixtoEsp).

Necesidad del recálculo de la información sensorial asociada a los sensores no lineales cuando llega al algoritmo información retardada proveniente de cualquier sensor

Para analizar la necesidad del recálculo de la información sensorial asociada a los sensores no lineales, ejecutaremos las dos versiones del algoritmo C3-3, sobre el caso de medidas desordenadas (b) del grupo experimental V. En ninguno de los dos casos se utilizará la etapa de validación, para evitar la influencia que tiene dicha etapa en los resultados obtenidos por cada uno de los algoritmos. Los resultados obtenidos para el mismo ejemplo por los dos algoritmos se encuentran recogidos en las gráficas de la figura C4-24. En la gráfica (a), donde se representan los valores estimados del estado por IFAsynMixto e IFAsynMixtoEsp, no se puede observar a simple vista la discrepancia existente entre los valores estimados por cada versión del algoritmo C3-3, que ha sido representada en la gráfica (c) de la misma figura. En las gráficas (b) y (d) se representan, respectivamente, el valor de la varianza estimada para cada estado por cada versión del algoritmo, y la discrepancia existente entre las varianzas estimadas. Por último en la gráfica (e) se representa la importancia de la discrepancia existente entre los estados obtenidos, medida como la razón entre el valor absoluto de dicha discrepancia y el valor de la desviación estándar asociada al mismo estado e instante de muestreo.

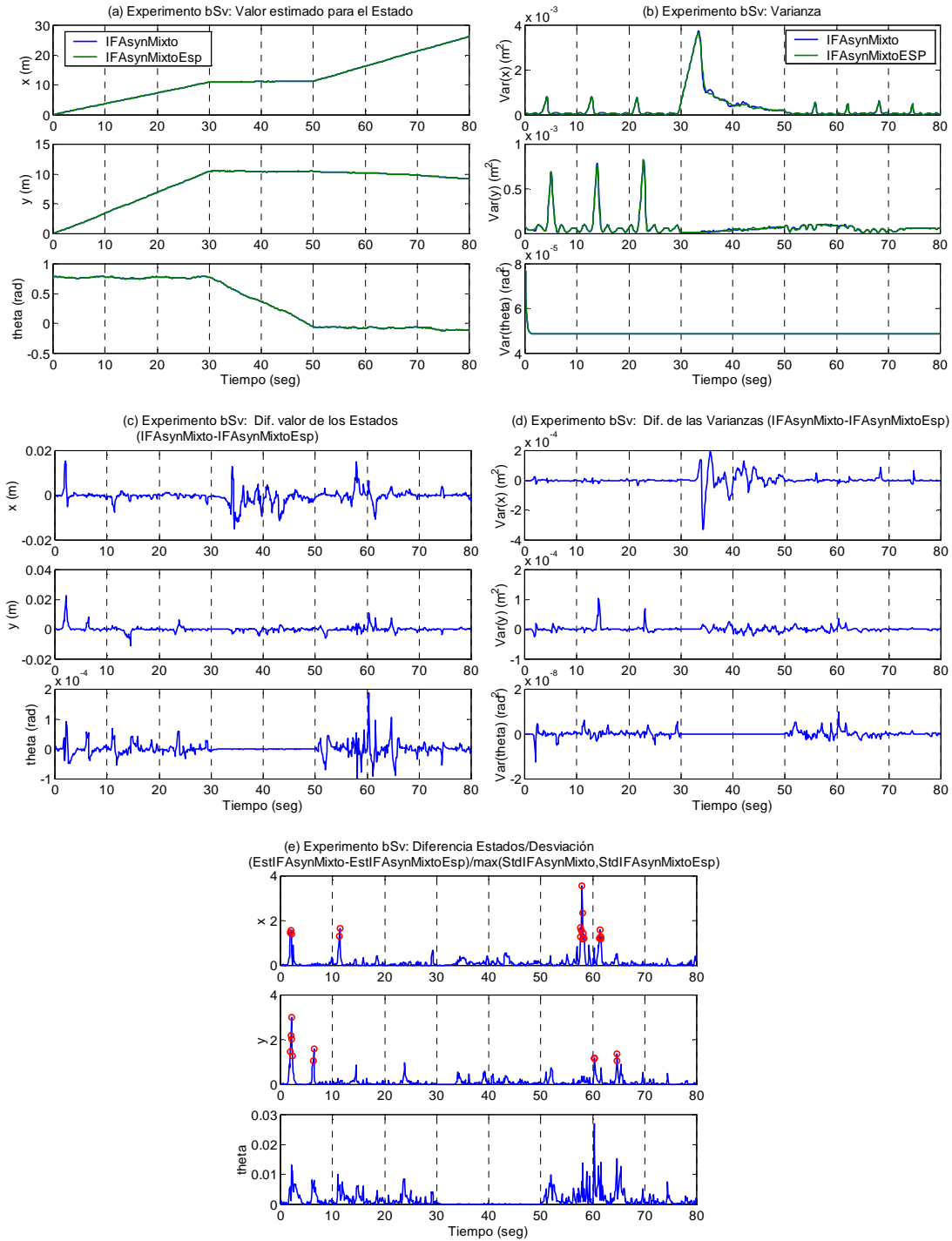


Figura C4-24. Resultados habituales del algoritmo C3-3 (IFAsynMixto) y de una versión modificada del mismo (IFAsynMixtoEsp, que no recalcula la información sensorial asociada a los sensores no lineales cuando llega información sensorial retardada), cuando ambos algoritmos son ejecutados sin etapa de validación sobre el experimento b del grupo experimental V. En la gráfica (a) se muestran los valores del estado estimados por los dos algoritmos y en la gráfica (b) las varianzas obtenidas. En la gráfica (c) se recoge la discrepancia entre los valores del estado estimados por los dos algoritmos y en la gráfica (d) la diferencia de las varianzas. En la gráfica (e) se muestra la importancia de la discrepancia entre los valores estimados del estado por cada versión del algoritmo relativa al valor de la desviación estándar máxima calculada.

En este caso, la discrepancia existente entre el valor del estado estimado por cada algoritmo se puede considerar significativa, ya que en algunos puntos, representados en rojo en la gráfica (e) de la figura C4-24, supera el valor de su correspondiente desviación estándar, e incluso la duplica. También se puede observar, comparando las gráficas (d) de la figuras C4-21 (sistema con funciones no lineales suaves) y figura C4-24 (sistema con funciones no lineales abruptas) que la diferencia de varianzas para este ejemplo es de un orden de magnitud superior que para el caso del sistema con funciones no lineales suaves. Además, para el último sistema ejemplo, se ha comprobado que la discrepancia media existente entre el valor del estado real y el estimado por el algoritmo IFAsynMixto es menor que la discrepancia existente entre el valor del estado real y el estimado por el algoritmo IFAsynMixtoEsp.

Por último, es importante destacar que los resultados recogidos en la figura C4-24 representan un caso típico de las múltiples simulaciones realizadas. En algunas de las mismas, los resultados han sido mucho peores, llegando a obtener discrepancias superiores a 1000 veces el valor de la desviación estándar. En las gráficas de la siguiente figura se representa parte de los resultados obtenidos, tras la generación aleatoria de una nueva trayectoria y las medidas de los sensores, para uno de los casos peores. En ella se puede ver como la discrepancia entre los estados es del orden de metros, y la razón de dicha discrepancia al valor de la desviación estándar es superior a las 100 unidades para los estados x e y .

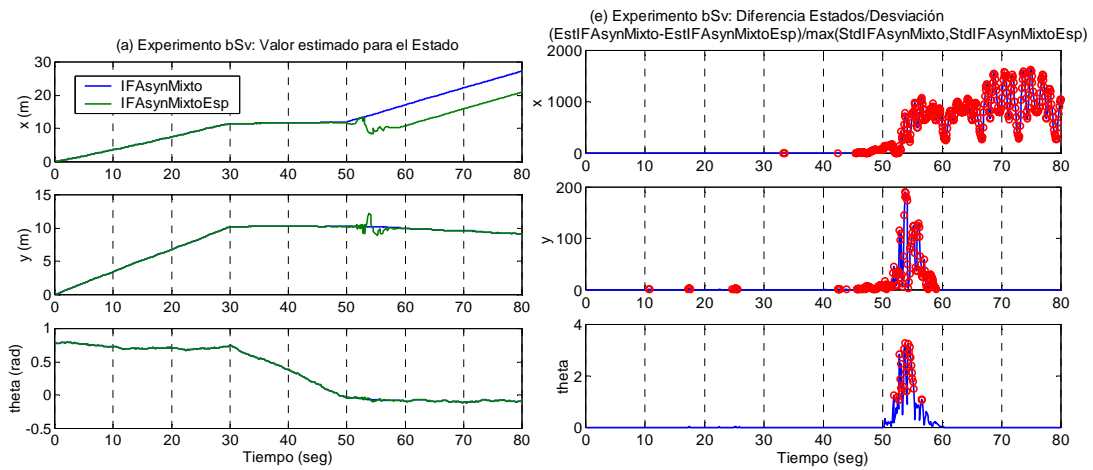


Figura C4-25. Resultados para un caso particular de la ejecución del algoritmo C3-3 (IFAsynMixto) y de una versión modificada del mismo (IFAsynMixtoEsp, que no recalcula la información sensorial asociada a los sensores no lineales cuando llega información sensorial retardada), cuando ambos algoritmos son ejecutados sin etapa de validación. En la gráfica (a) se representan los valores del estado estimado por los dos algoritmos y en la gráfica (b) se representa la importancia de la discrepancia entre los valores estimados del estado por cada versión del algoritmo relativa al valor de la desviación estándar máxima calculada.

Para interpretar estos resultados hay que tener en cuenta que las funciones no lineales elegidas para el sensor2 y sensor3 presentan cambios abruptos periódicamente, y que según aumenta el Δx , el error cometido en la información sensorial no recalculada por el algoritmo IFAsynMixtoEsp puede

ser mayor, y si se acumula, aumentará el valor del Δx del siguiente instante de muestreo, y así sucesivamente. Este hecho es el causante de que según los valores generados aleatoriamente la discrepancia entre los valores obtenidos por los dos algoritmos para el ejemplo con funciones no lineales abruptas produzca resultados similares, aunque con una discrepancia significativa, como en el caso de la primera simulación presentada, o muy diferentes, como en el caso de la segunda.

Por lo tanto, se puede concluir que el recálculo de la información sensorial asociada a los sensores no lineales cuando el algoritmo C3-3 recibe medidas retardadas, es una operación necesaria para algunos sistemas, y que la falta de dicho cálculo puede introducir no solo errores no despreciables, sino que en algunos casos, errores muy significativos. La decisión sobre la realización de dicho recálculo, y por lo tanto sobre la versión del algoritmo C3-3 (IFAsynMixto o IFAsynMixtoEsp) que se utilice, dependerá de las características del sistema elegido, cuyos resultados deberán ser analizados en múltiples simulaciones, antes de elegir la versión modificada (IFAsynMixtoEsp) frente a la versión original (IFAsynMixto). A la hora de la elección, es conveniente tener siempre en cuenta, que salvo por la influencia de la etapa de validación en el orden de la medida en los resultados obtenidos por el algoritmo IFAsynMixto, éste obtiene los mismos valores que el filtro de Información original para las mismas medidas recibidas de forma ordenada.

Capítulo V

Utilización del Sistema de Fusión en una Sistema de Control en Tiempo Real para Robots Autónomos Móviles

En este capítulo se describe el sistema de control en tiempo en real para robots autónomos móviles sobre el que se comprueba el funcionamiento del algoritmo IFAsynMixto sobre un sistema real, y se presentan los resultados obtenidos por el mismo cuando la arquitectura de control elegida es utilizada sobre un robot real o para simular el comportamiento del mismo. El robot utilizado como banco de pruebas, descrito detalladamente en el apéndice III, ha sido diseñado y construido como parte integral de esta tesis como una plataforma multisensorial abierta, versátil, y fácilmente configurable.

El sistema de control del robot está formado por un conjunto de módulos, implementados como una arquitectura de control híbrida compuesta por múltiples tareas concurrentes que intercambian información a través de mensajes. Las propiedades más relevantes de la misma se presentan en [Lopez-Orozco1998b], [Lopez-Orozco1999] y [Besada-Portas2002b], entre las que cabe destacar un enfoque especialmente orientado al tratamiento de la información sensorial y una elevada adaptabilidad que permite utilizar la información extraída para diferentes objetivos, incrementar y modificar la funcionalidad del sistema fácilmente, y reutilizar la arquitectura de control para distintas plataformas robóticas.

Las características finales de la arquitectura de control se encuentran íntimamente ligadas a la metodología elegida para definir la funcionalidad del sistema y realizar su implementación. Para llevar a cabo estas tareas se ha seleccionado el paradigma de programación gráfico orientado a objetos para el modelado de sistemas de tiempo real ROOM (Real-Time Object-Orient Modeling) [Selic1995], debido a su gran potencia, versatilidad y facilidad de uso. Esta metodología es soportada por la herramienta de distribución libre *EdROOM* ([Rodriguez-Polo2003a]), desarrollada en el Departamento de Arquitectura de Computadores y Automática de la Universidad Complutense de Madrid y caracterizada por contemplar las tres etapas del ciclo diseño-implementación-análisis. En el apéndice IV se recogen las principales características de la metodología ROOM y de la herramienta *EdROOM*, y se hace hincapié en aquellas propiedades que hacen que tanto la metodología como la herramienta sirvan para implementar los diferentes tipos de arquitecturas robóticas existentes.

La incorporación del algoritmo IFAsynMixto a la arquitectura robótica constituye únicamente la etapa final del trabajo que se ha realizado en esta tesis sobre el sistema de control de alto nivel del robot, ya que la versión original de la arquitectura, presentada en [Lopez-Orozco1998b] y [Lopez-Orozco1999], ha sido modificada para mejorar sus prestaciones y optimizar su rendimiento; y la funcionalidad de sus módulos y las relaciones existentes entre los mismos especificados gráficamente por medio de la herramienta *EdROOM*.

La división establecida entre los diferentes módulos de la arquitectura permite reutilizar la mayor parte de los módulos, los de control de alto nivel y de fusión multisensorial, para estimar el estado del entorno y la posición de diferentes robots y realizar un control autónomo del mismo en base a la información extraída de los sistemas de fusión. El tipo de control o los algoritmos de fusión también pueden ser modificados, en aquellos casos en los que se considere oportuno.

En este capítulo se recogen los resultados obtenidos por el sistema de fusión multisensorial, y especialmente aquellos obtenidos por el algoritmo IFAsynMixto, propuesto en el capítulo III, cuando al módulo encargado de interactuar directamente con el robot se le incorpora la funcionalidad necesaria para controlar al robot descrito en el apéndice III de la tesis, o simular el comportamiento de sus sensores o actuadores. De esta forma, el mismo sistema de control, salvo por las funciones utilizadas para simular o interactuar con el robot, es probado bajo simulación o en un sistema real. Las ecuaciones de modelo de movimiento del robot y de medida de los sensores, así como la adaptación del algoritmo IFAsynMixto para asegurar el correcto funcionamiento del algoritmo sobre el robot elegido, se presentan en el apéndice V.

Los resultados obtenidos mediante simulación permiten analizar el funcionamiento de la arquitectura de control junto con el algoritmo IFAsynMixto cuando se utilizan las mismas ecuaciones para simular el comportamiento de los sensores y los actuadores del robot, y para estimar su posición dentro del filtro. Este hecho nos permite comprobar el funcionamiento del algoritmo en una situación óptima, al coincidir los modelos del sistema y de los sensores, con sus comportamientos simulados. Además, la simulación favorece la utilización simultánea de mayor número de sensores, al permitir que se consideren despreciables los tiempos necesarios para interactuar con el robot y sus sensores. Es decir, no restringe el análisis del funcionamiento de los algoritmos a las limitaciones inherentes al sistema sobre el que serán probados en los experimentos reales. Los resultados simulados de este capítulo son diferentes a los resultados obtenidos en las simulaciones realizadas en el capítulo anterior, no sólo por la utilización de unas ecuaciones de modelo de movimiento y de medida diferentes, sino porque en este capítulo se realiza un control en lazo cerrado del sistema sobre el que se analiza el comportamiento del algoritmo.

Los resultados obtenidos en las pruebas reales permiten estudiar el funcionamiento de la arquitectura de control junto con el algoritmo IFAsynMixto sobre un sistema real. En este caso, los resultados obtenidos dependen de la completitud y corrección de los modelos de movimiento y de medida utilizados para estimar la posición del robot. Por lo tanto, en las pruebas reales no sólo se analiza la bondad del algoritmo, sino que además se comprueba la exactitud de los modelos propuestos en el apéndice V para el robot descrito en el apéndice III.

Este capítulo se encuentra dividido en dos secciones. En la primera se describe la arquitectura de control en la que se incluye el algoritmo IFAsynMixto, haciendo un especial hincapié en la descripción del sistema de fusión multisensorial de la misma, mientras que en la segunda se recogen los resultados obtenidos por los algoritmos de fusión, bajo simulación y cuando la arquitectura de control interactúa con el robot descrito en el apéndice III.

C5.1. Incorporación del algoritmo IFAsynMixto a una arquitectura robótica de control

El sistema real elegido para probar el comportamiento del filtro de Información Mixto para la asimilación de medidas retardadas y desordenadas (IFAsynMixto) es un robot autónomo móvil, diseñado y construido como parte integral de esta tesis para poder disponer de una plataforma multisensorial abierta, versátil, y fácilmente configurable. Una descripción del mismo, en la que se analiza de forma detallada el funcionamiento de los distintos dispositivos sensoriales y de control presentes en el robot, es recogida en el apéndice III.

El robot es controlado por un complejo sistema de tareas, responsables de (1) planificar, supervisar y llevar a cabo las labores para las que éste ha sido diseñado, y (2) asegurar su autonomía mediante la implementación de una cierta capacidad de reacción ante los cambios imprevistos de su entorno. La organización y relación existente entre las tareas en las que se descompone la funcionalidad del sistema (también denominadas módulos en [Alami1998] y [Fleury1994], agentes inteligentes en [Russell1995] y [Müller1996], agentes reactivos en [Brooks1986], o comportamientos y esquemas motores en [Arkin1989]) se conoce como arquitectura de control y ha dado lugar a dos tendencias de diseño opuestas. Por una parte, las arquitecturas deliberativas ([Elfes1986], [Albus1987] y [Simmons1994]) con un enfoque centralizado y jerárquico que utiliza representaciones simbólicas del mundo para realizar razonamientos complejos, y por otra, las arquitecturas reactivas ([Brooks1986], [Payton1991] y [Gachet1994]), con un enfoque descentralizado en tareas que compiten simultáneamente por realizar las tres etapas del lazo de control (sensorización, decisión y actuación). El grado de abstracción logrado por el primer tipo de arquitecturas permite al sistema tener una visión más completa del estado del robot y de su entorno, y predecir el resultado de su comportamiento y el de otros objetos, a costa de penalizar la velocidad de respuesta del sistema frente a cambios y sucesos que no ha sido capaz de predecir. La reactividad del segundo tipo asegura velocidades de respuesta elevadas pero dificulta la definición flexible de las tareas del robot y la consecución de objetivos globales.

A medio camino entre ambas filosofías se han desarrollado múltiples arquitecturas híbridas ([Arkin1992], [Gat1992], [Firby1994], [Chatilla1995], [Bonasso1997], [Barber2001], etc.) con el objeto de aprovechar las ventajas y superar las limitaciones de los dos tipos de arquitecturas extremas. La arquitectura en la que se incorpora el algoritmo IFAsynMixto, cuyas propiedades más relevantes se presentan en [Lopez-Orozco1998b], [Lopez-Orozco1999] y [Besada-Portas2002b] pertenece a este último tipo, y se encuentra caracterizada por un enfoque especialmente orientado al tratamiento de la información sensorial y una elevada adaptabilidad que permite utilizar la

información extraída para diferentes objetivos, incrementar y modificar la funcionalidad del sistema fácilmente, y reutilizar la arquitectura de control para distintas plataformas robóticas.

Las características finales de la arquitectura de control no sólo dependen de las relaciones existentes entre los módulos que la constituyen, sino que además se encuentran íntimamente ligadas a la metodología elegida para definir su funcionalidad y realizar la implementación del sistema. Entre las diferentes metodologías existentes se ha elegido el paradigma de programación gráfico orientado a objetos para el modelado de sistemas de tiempo real ROOM (Real-Time Object-Orient Modeling) [Selic1995], debido a su gran potencia, versatilidad y facilidad de uso. Esta metodología es soportada por la herramienta de distribución libre *EdROOM* ([Rodriguez-Polo2003a]), desarrollada en el Departamento de Arquitectura de Computadores y Automática de la Universidad Complutense de Madrid y caracterizada por contemplar las tres etapas del ciclo diseño-implementación-análisis: (1) el diseño gráfico de la estructura y comportamiento del sistema, (2) el análisis de su funcionamiento, y (3) la implementación del sistema modelado en un programa C++, que una vez compilado tiene que ser enlazado con una librería de servicios *EdROOM* que sirve de interfaz entre el sistema operativo sobre el que se desea ejecutar el programa y el código C++ generado. En el apéndice IV se recogen las principales características de la metodología ROOM y de la herramienta *EdROOM*, haciendo al final del mismo un especial hincapié en aquellas propiedades que hacen que tanto la metodología como la herramienta sirvan para implementar los diferentes tipos de arquitecturas robóticas existentes.

Es importante poner de manifiesto que la incorporación del algoritmo IFAsynMixto a la arquitectura robótica constituye únicamente la etapa final del trabajo que se ha realizado en esta tesis sobre el sistema de control de alto nivel del robot. La arquitectura original presentada en [Lopez-Orozco1998b] y [Lopez-Orozco1999] ha sido modificada para mejorar sus prestaciones y optimizar su rendimiento, y la funcionalidad de sus módulos y las relaciones existentes entre los mismos especificados gráficamente por medio de la herramienta *EdROOM*. Además, ha sido necesario implementar la librería de servicios *EdROOM* para Windows 2000, ya que ésta no existía y éste es el sistema operativo elegido para ejecutar el programa de control del robot, cuyo código C++ es generado automáticamente por la herramienta de acuerdo con la nueva especificación gráfica del sistema. De esta forma, no solo se ha utilizado *EdROOM* como herramienta de diseño, sino que se ha ampliado el conjunto de sistemas operativos sobre los que se podrán ejecutar otros sistemas cuya estructura y comportamiento se encuentren definidos gráficamente por medio de la herramienta *EdROOM*. Las características de la librería de servicios *EdROOM* implementada para Windows 2000 también se presentan en el apéndice IV.

En los siguientes apartados de esta sección se realiza un análisis de los tipos de arquitecturas de control existentes y de las características deseables en las metodologías de implementación, y se

presenta la organización y los módulos de la arquitectura utilizada para el control de nuestro robot durante los experimentos con el algoritmo IFAsynMixto.

C5.1.1. Arquitecturas de control existentes

Una técnica básica para implementar la funcionalidad de cualquier sistema complejo consiste en dividir ésta en un conjunto de tareas o módulos, y dotar a cada uno de los mismos con una parte de la funcionalidad total. En los sistemas de control para robots autónomos móviles, cuyo ciclo de control está formado por las tres etapas básicas (sensorización, decisión y actuación) representadas en la figura C5-1, la división en tareas da lugar a tres tipos de arquitecturas: deliberativas, híbridas y reactivas. Éstas se diferencian por la organización y relación existente entre las tareas que las constituyen y las tres etapas del ciclo de control, el tipo de razonamiento que realizan, la velocidad de respuesta del sistema ante cambios imprevistos, etc.

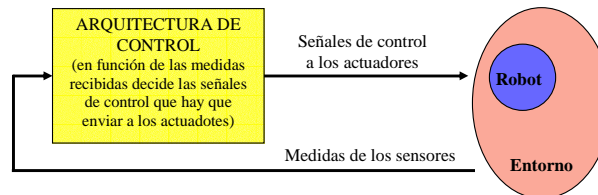


Figura C5-1. Lazo para el control de robots autónomos móviles.

▪ Arquitecturas deliberativas o jerárquicas

En las arquitecturas deliberativas (también conocidas bajo el nombre de arquitecturas jerárquicas y centralizadas y que se encuentran implementadas en los sistemas descritos en [Elfes1986], [Albus1987] y [Simmons1994]) la funcionalidad del sistema se segmenta en un conjunto de tareas responsables de alguna de las partes de las diferentes etapas del lazo de control. Éstas son organizadas consecutivamente de forma que se establece un flujo unidireccional de la información del sistema dentro de la arquitectura.

En la figura C5-2 se representa una arquitectura de control deliberativa formada por cuatro módulos diferentes, encargados de realizar las labores de percepción (interpretación de la información sensorial, integración y fusión de la misma), planificación (de trayectorias, esquiva de objetos), supervisión (del correcto funcionamiento del robot) y control (de las actividades realizadas por la plataforma móvil). En la misma se puede observar como dentro de este tipo de arquitectura de control el lazo se cierra con un flujo unidireccional de la información a través de sus diferentes tareas.

La información procedente de los sensores del robot es procesada según atraviesa los diferentes módulos de la arquitectura mediante razonamiento deliberativo, realizando deducciones lógicas y

predicciones sobre el modelo del mundo utilizado y el comportamiento deseado, y retorna a los actuadores del robot bajo la forma de señal de control. La jerarquía establecida en este tipo de arquitectura facilita la estructuración de formas de razonamiento complejos compuestos por diferentes niveles de abstracción y el análisis del funcionamiento del sistema determinado por el flujo unidireccional de la información a través de los módulos que lo constituyen.

Como contrapartida, el flujo unidireccional de la información hace necesario una implementación completa de la funcionalidad asignada a cada tarea y el mantenimiento de la interfaz existente entre dos módulos consecutivos, y penaliza con un tiempo de respuesta elevado los cambios imprevistos del entorno observados por los sensores. El incremento de la funcionalidad del sistema no es una labor sencilla, ya que hay que modificar el comportamiento de uno o varios módulos, poniendo especial cuidado en el mantenimiento de la consistencia del flujo de la información a través de las tareas que constituyen la arquitectura.

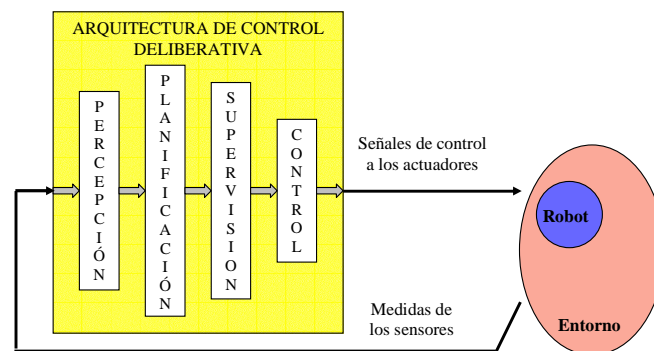


Figura C5-2. Segmentación de tareas en una arquitectura de control deliberativa.

▪ **Arquitecturas reactivas**

En las arquitecturas reactivas, también conocidas como arquitecturas distribuidas y que se encuentran implementadas en los sistemas descritos en [Brooks1986], [Payton1991] y [Gachet1994], la funcionalidad del sistema se distribuye en un conjunto de capas independientes entre sí, todas ellas responsables de realizar algún tipo de razonamiento a partir de las medidas proporcionadas por los sensores del que se puede concluir algún tipo de señal de control aplicable al robot, y por lo tanto, todas ellas encargadas de implementar funciones asociadas a las tres etapas básicas del lazo de control (sensorización, decisión y actuación).

En la figura C5-3 se representa una arquitectura de control reactiva formada por un conjunto de ocho módulos diferentes, encargados de diferentes niveles de razonamiento. En la misma se puede observar como dentro de este tipo de arquitectura se establecen un flujo de información y lazo de control multicapa.

La información procedente de los sensores es procesada por cada uno de los módulos de la arquitectura y la consecuencia directa de dicho procesamiento es la generación de órdenes básicas de control para los actuadores. Con esta organización se pueden conseguir sistemas de elevada reactividad, donde todas las tareas realizan un procesamiento simultáneo de la información sensorial y los razonamientos de los niveles inferiores proporcionan respuestas rápidas ante sucesos imprevistos. Además, se puede incrementar fácilmente la funcionalidad del sistema a través de la incorporación de nuevos módulos, y depurar los comportamientos asociados a cada tarea del sistema de forma independiente.

Como contrapartida, el sistema no tiene una visión global del mundo, ya que cada tarea realiza un razonamiento independiente del razonamiento realizado en las tareas restantes. Esto impide la utilización de modelos complejos, completos y precisos, sobre los que realizar deducciones y predicciones de alto nivel para alcanzar objetivos globales. Además, las acciones básicas propuestas simultáneamente por diferentes módulos tienen que ser coordinadas por medio de algún mecanismo, como puede ser la inhibición de medidas y supresión de acciones propuesta en [Brooks1986] o la fusión adaptativa de acciones de [Gachet1994]. El análisis de comportamiento final del sistema es una labor compleja, ya que no existe un flujo único de información a través de la arquitectura, y aunque sea fácil incorporar nuevos módulos, la flexibilidad y adaptabilidad real de la arquitectura puede verse degradada debido a la dificultad de conseguir el comportamiento global deseado a través de la coordinación de las acciones básicas originadas en cada tarea.

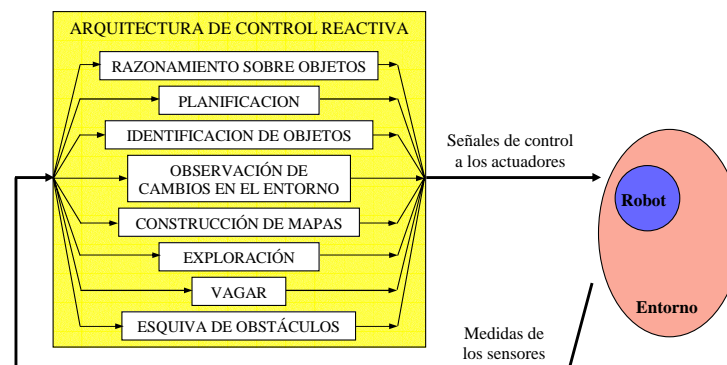


Figura C5-3. Estructura multicapa de una arquitectura de control reactiva

▪ **Arquitecturas Híbridas**

Como una solución intermedia y natural ante las dos posibilidades extremas que representan las arquitecturas deliberativas y reactivas, se ha desarrollado un amplio conjunto de arquitecturas híbridas ([Arkin1992], [Gat1992], [Firby1994], [Chatilla1995], Bonasso1997], [Barber2001], etc.) con el objeto de aprovechar las ventajas y superar las limitaciones de cada arquitectura extrema. Estas arquitecturas realizan razonamiento simbólico sobre un modelo del mundo que es

actualizado con la información proporcionada por los sensores, y mantienen la velocidad de respuesta y flexibilidad de los sistemas reactivos al permitir la ejecución concurrente de diferentes niveles dentro del lazo de control.

Habitualmente están formadas por dos niveles diferentes, uno reactivo que asegura el funcionamiento del robot a corto plazo y otro deliberativo encargado de realizar la planificación de las tareas y el cálculo de las trayectorias a lo largo plazo, y que se encuentran coordinados por medio de alguna estrategia. En [Lyons1992] se presentan tres posibles relaciones entre los dos niveles (integración jerárquica de los niveles deliberativos y reactivos, guiado de las acciones del nivel reactivo a través de las decisiones tomadas por el nivel deliberativo, y concurrencia entre ambos niveles) mientras que en [Arkin1998] se proponen cuatro (selección por el nivel deliberativo de los comportamientos y parámetros utilizados por el nivel reactivo, asesoramientos del planificador al nivel reactivo sobre los cambios que debe realizar respecto al control del robot, adaptación del comportamiento del nivel reactivo de acuerdo con los cambios observados por el nivel deliberativo, y aplazamiento de las decisiones tomadas por el nivel deliberativo hasta que estas sean realmente necesarias).

La arquitectura en la que se incorpora el algoritmo IFAsynMixto ([Lopez-Orozco1998b], [Besada-Portas2002b]) pertenece a este último tipo y durante la división de la funcionalidad del sistema en diferentes tareas se ha procurado obtener una organización especialmente orientada al tratamiento de la información sensorial por medio de técnicas de fusión multisensorial, donde la información extraída pueda ser utilizada para la consecución de diferentes objetivos globales. La arquitectura obtenida se caracteriza por su versatilidad y reusabilidad, características que se traducen en facilidad para incrementar y modificar su funcionalidad y la posibilidad de utilizar el sistema de control para diferentes objetivos y/o plataformas robóticas. En la siguiente figura se representan un esquema general de la arquitectura híbrida utilizada.

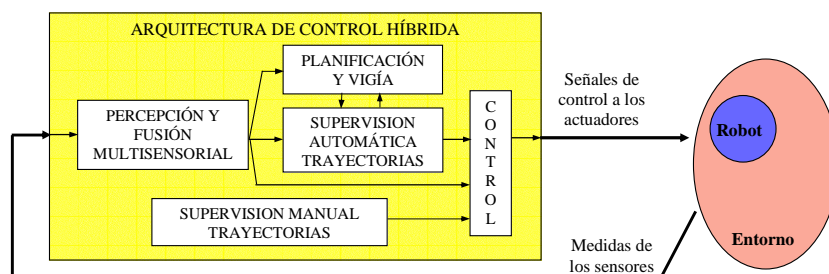


Figura C5-4. Esquema general de la arquitectura híbrida en la que se incorpora el algoritmo IFAsynMixto.

C5.1.2. Metodología de implementación

La versatilidad de la arquitectura elegida no depende únicamente de la organización de tareas propuesta, sino que sus características finales se encuentran íntimamente ligadas con la metodología elegida para definir la funcionalidad de las diferentes tareas y las relaciones existentes entre las mismas. Características a tener en cuenta durante la elección de la misma son que ésta permita especificaciones claras y de fácil comprensión, preferentemente visual, y los mecanismos de los que dispone para asegurar, especialmente en los casos de las arquitecturas reactivas e híbridas, el cumplimiento de las restricciones temporales existentes a la hora de realizar un control en tiempo real ([Laplante1993], [Bennet1994]) del robot.

Aunque siempre es posible diseñar una metodología especialmente orientada a la arquitectura propuesta, resulta más sencillo utilizar alguna técnica genérica de diseño y análisis para sistemas de tiempo real. Tras el estudio de algunas de las metodologías orientadas a objeto existentes mientras se desarrollaba la primera versión de la arquitectura de control, presentada en [Lopez-Orozco1998b] y [Lopez-Orozco1999], se optó por la utilización de ROOM ([Selic1995]) debido a su gran potencia, versatilidad y facilidad de uso. Entre las características de este formalismo destacan: (1) una especificación gráfica de la estructura y comportamiento del sistema utilizable tanto en la etapa de diseño como en la de documentación; (2) el ser específica del diseño e implementación de sistemas en tiempo real debido a la concurrencia y reactividad de sus módulos; y (3) su elevado nivel de abstracción y recursividad como consecuencia del tratamiento de los módulos del sistema como máquinas lógicas independientes que funcionan de forma distribuida y concurrente, y al uso de tantos niveles de refinamiento como se considere oportuno en la definición de la estructura y comportamiento de cada módulo.

Todas estas características permiten implementar diferentes tipos de arquitecturas robóticas, si bien ROOM conduce al usuario de la metodología de forma natural a la implementación de arquitecturas híbridas, debido a la combinación de reactividad y capacidad de abstracción que caracterizan a la herramienta. Aún más, la definición gráfica de todo el sistema, desde la estructura hasta su comportamiento, y la existencia de tantos niveles de encapsulamiento como se considere oportuno favorece el diseño de arquitecturas robóticas fácilmente adaptables.

La metodología elegida es soportada por la herramienta de distribución libre *EdROOM* ([Rodriguez-Polo2003a]) que permite al usuario: (1) realizar el diseño gráfico de la estructura y comportamiento del sistema, (2) implementar en sistema modelado como un programa C++ y (3) analizar el funcionamiento del programa compilado. En la definición del comportamiento del sistema se utilizan diagramas de estado, cuyas transiciones son disparadas a la recepción de mensajes, de datos y/o temporización, y cuya funcionalidad se implementa mediante funciones C++ definidas directamente con la herramienta o en librerías de comportamiento externas. El

código C++ asociado a la especificación gráfica es generado de forma automática por la herramienta, incorporando en el mismo las funciones especificadas en la herramienta. Para obtener un programa ejecutable, el código generado tiene que ser compilado y enlazado con las librerías de comportamiento externo y con una librería de servicios *EdROOM*, en la que se encuentran definidas un conjunto de funciones que sirven como interfaz entre el código C++ generado y el sistema operativo sobre el que se ejecutará el programa.

La primera versión de la arquitectura robótica ([Lopez-Orozco1998b] y [Lopez-Orozco1999]) fue implementada siguiendo la metodología ROOM pero sin utilizar el editor gráfico (en el que se especifica la estructura y el comportamiento de los módulos del sistema) ni el generador automático de código C++ de *EdROOM*, ya que estas dos partes de la herramienta se encontraban en fase de desarrollo. Este hecho no impidió la utilización de la metodología aunque si dificultó su uso, ya que estas dos partes de la herramienta eliminan el vacío existente entre la etapa de diseño y la de implementación final del sistema. Para la versión de la arquitectura presentada en [Besada-Portas2002b], en la que posteriormente se ha incorporado el algoritmo IFAsynMixto, el editor gráfico y el generador de código de *EdROOM* ya se encontraban disponibles, lo que permitió realizar la definición de la estructura y el comportamiento de la nueva versión por medio del editor, y obtener automáticamente su código C++. El esfuerzo inicial realizado para redefinir totalmente la arquitectura de control del robot mediante *EdROOM* se ve compensado por las nuevas facilidades de diseño e implementación proporcionadas por la herramienta. Aún más, el proceso de evolución de la arquitectura robótica hacía su versión actual, presentada a continuación, se ha simplificado por el uso de la herramienta, al poder introducir sistemáticamente las modificaciones oportunas.

Buena parte de las modificaciones introducidas en la última versión de la arquitectura de control se han realizado con el objeto de buscar un equilibrio entre el grado de abstracción logrado por la multiplicidad de niveles soportada por ROOM y la complejidad real del sistema final, implementado mediante el generador de código de *EdROOM*. Los tres métodos de refinamiento disponibles en ROOM (uno en la definición de la estructura donde un módulo puede incluir en su definición tantos módulos como se considere oportuno; y dos en la definición del comportamiento, a través de la multiplicidad de niveles en la definición de los diagramas de estados y el uso de clases para especificar la información y funcionalidad asociada a las variables) significan complejidades diferentes en la implementación final:

- Cada módulo es implementado como una tarea diferente e independiente, con una funcionalidad propia, que será ejecutada de forma concurrente con el resto de las tareas del sistema. Así pues, el uso de múltiples niveles en la definición de la estructura provoca la necesidad de crear una tarea diferente por cada módulo existente, hecho que incrementa la complejidad del sistema, aun en los casos que el módulo que encapsula a los módulos de un

nivel inferior no tenga una funcionalidad real. Además, el intercambio de información entre módulos únicamente se realiza a través del paso de mensajes, hecho que sobrecargará computacionalmente al sistema cuando dos módulos trabajen sobre el mismo tipo de información y necesiten intercambiarla continuamente.

- El comportamiento de cada módulo se implementa mediante un diagrama de estados, y cada estado puede estar formado asimismo por un diagrama de estados. Asociado a cada nivel (contexto) se encuentran las variables y funciones definidas en el mismo. De esta forma un estado puede encapsular el comportamiento de un conjunto de estados, junto con las variables y funciones de los mismos. Las variables y funciones de un nivel superior son accesibles desde cualquier nivel inferior, hecho que no se da en el caso contrario. El encapsulamiento de contextos incrementa la complejidad de las clases definidas por *EdROOM* en el código C++ asociado al modelo y del proceso de elección de la transición disparada por un determinado evento, hechos que afectarán al rendimiento final del sistema.
- El uso de un paradigma orientado a objetos permite estructurar y encapsular la definición de las variables y funciones de un contexto en clases, hecho que simplificará el tratamiento de la elevada complejidad del sistema de control en tiempo real.

Por último, es importante resaltar dos características de *EdROOM* que han aumentado la versatilidad y portabilidad de nuestro sistema de control:

- Por una parte, las funciones utilizadas en los diagramas de estado de los módulos de la arquitectura para especificar el comportamiento del sistema pueden encontrarse definidas en librerías de comportamiento externas a *EdROOM*. Este hecho, además de ayudar al usuario a depurar el funcionamiento del código externo antes de utilizarlo como parte de la arquitectura de control, permite construir modelos de control generales y reutilizables en distintos robots, donde la funcionalidad directamente relacionada con cada robot se encuentre definida bajo un conjunto de clases único e implementada en un conjunto de librerías externas diferente para cada caso. Del mismo modo que se pueden tener librerías externas para diferentes robots, es posible implementar para el mismo robot una librería que se encargue de interactuar con sus sensores y actuadores y otra de simular el comportamiento de estos dispositivos. Este hecho permite al usuario utilizar una única arquitectura de control para observar el comportamiento real o simulado del sistema, con tan sólo enlazar el código C++ generado automáticamente con la librería de control real o la librería de simulación. Para obtener los resultados experimentales (mediante simulación y sobre el robot real) recogidos en la segunda parte de este capítulo, se hace uso de esta característica de *EdROOM*.
- Por otra, el código C++ generado de forma automática por la herramienta puede ser independiente del sistema operativo sobre el que se ejecuta el programa final debido a que

las funciones de los servicios de ROOM directamente relacionados con el sistema operativo se encuentran implementadas en las librerías de servicios *EdROOM*. De esta forma, las posibles dependencias entre el código C++ generado y el sistema operativo serán únicamente originadas por las funciones que el usuario defina para implementar la funcionalidad de los módulos del sistema, hecho que no reducirá la generalidad del modelo especificado mediante *EdROOM* siempre y cuando estas funciones se encuentren recogidas en librerías de comportamiento externo. La portabilidad del modelo *EdROOM* se encuentra soportada por la existencia de librerías de servicios *EdROOM* para diferentes sistemas operativos en tiempo real, como son RTKernel (www.on-time.com), CMX (www.cmx.com) y RTAI (www.airi.plimi.it/~rtai). El número de librerías de servicios implementadas puede ser incrementado con nuevas versiones para otros sistemas operativos siempre y cuando se mantenga la interfaz externa de todas las clases de la librería de servicios. Las pruebas experimentales de la arquitectura de control en la que se ha incorporado el algoritmo IFAsynMixto tenían que ser realizadas sobre el sistema operativo Windows 2000, ya que una parte de la funcionalidad del sistema, implementada en una librería de comportamiento externo, utilizaba una librería dependiente de dicho sistema operativo. Por este motivo, ha sido necesario desarrollar la librería de servicios *EdROOM* para Windows 2000, que a pesar de no ser un sistema operativo de tiempo real, permite modelar sistemas con restricciones temporales suaves. Esta librería, que constituye una parte del trabajo realizado durante el desarrollo de esta tesis, puede ser utilizada para la obtención de los programas ejecutables asociados a otros sistemas cuya estructura y comportamiento se encuentren definidos siguiendo la metodología ROOM por medio de la herramienta *EdROOM*.

En el apéndice IV se realiza una breve descripción de la metodología ROOM, en la que se presentan los elementos sobre los que ésta se sustenta y que han sido utilizados para implementar la arquitectura de control propuesta para el robot. Además, se expone el funcionamiento de las componentes principales de la herramienta de diseño *EdROOM*: el editor gráfico, el generador automático de código C++ y la librería de servicios *EdROOM*. De forma más detallada se presentan las características y funciones de Windows 2000 sobre las que se fundamenta la librería de servicios diseñada para este sistema operativo. Por último se ponen de manifiesto aquellas propiedades de la metodología ROOM y de la herramienta *EdROOM* que favorecen la implementación de arquitecturas robóticas versátiles, abiertas y reutilizables.

C5.1.3. Descripción de la arquitectura robótica

Tras el análisis de los tipos de arquitectura de control de robots existentes y de la metodología y herramienta utilizadas para la implementación de nuestra arquitectura realizaremos una descripción

de su estado actual. Tanto en la especificación de las relaciones existentes entre los módulos (definición estructural de ROOM) como en la especificación de la funcionalidad del sistema (definición de comportamientos de ROOM y utilización de un lenguaje orientado a objetos) se ha explotado la posibilidad de encapsulamiento de la metodología para conseguir un elevado nivel de abstracción.

En las figuras de esta sección se representan algunos esquemas estructurales y diagramas de estado de la versión actual de la arquitectura que han sido definidos por medio de la herramienta de diseño de *EdROOM*. Lo que en ellas se refiere a la definición en *EdROOM* (módulos, identificadores, puertos, conexiones, estados, etc.) se encuentra representado en blanco y negro. La información del sistema almacenada en el editor de ROOM es mucho más rica que la aquí representada. Por lo tanto, para sintetizar alguna información relevante no representada directamente en los gráficos extraídos del editor, se añaden elementos en los diferentes tipos de diagramas:

- En las figuras de los diagramas estructurales de esta sección se añaden a los esquemas generados por *EdROOM* elementos y leyendas explicativas, en diferentes colores.

En la siguiente figura se muestra la representación de la estructura del mismo módulo (el principal del sistema) de las dos formas diferentes: tal y como ha sido extraído del editor de *EdROOM* (a la izquierda), y con elementos (en este caso flechas que indican la dirección del flujo de la información) y leyendas (sobre un fondo amarillo, que indican el tipo de información intercambiada).

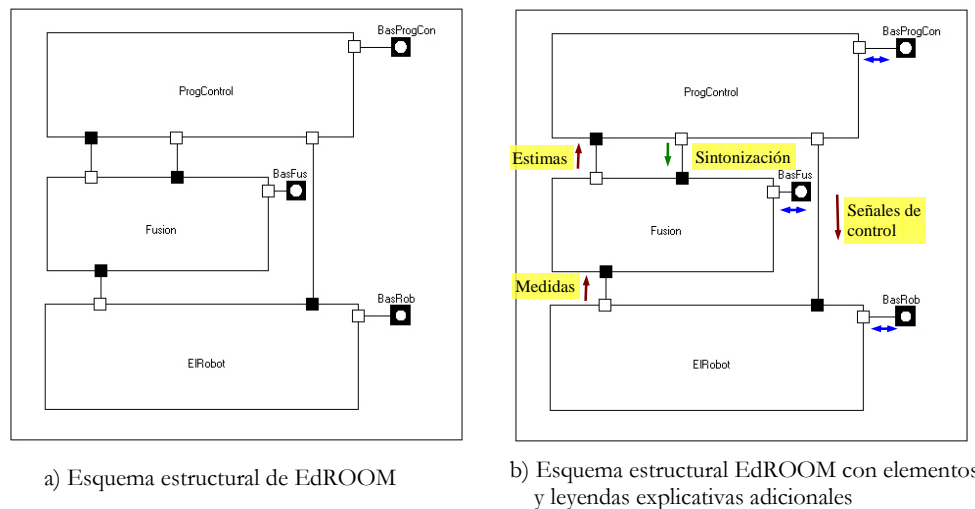


Figura C5-5. Diferencias entre un diagrama estructural en EdROOM y un esquema estructural con elementos explicativos adicionales.

- En las figuras de los diagramas de estado se colorean algunos estados y etiquetas de las transiciones para poner de manifiesto el tipo de funcionalidad asociado a los mismos.

En la siguiente figura se muestra el diagrama del mismo módulo (asociado al comportamiento del módulo *EIRobot*) de las dos formas diferentes: tal y como ha sido

extraído del editor de *EdROOM* (a la izquierda), y con los estados y transiciones asociadas a la inicialización y finalización de la tarea coloreadas en amarillo.

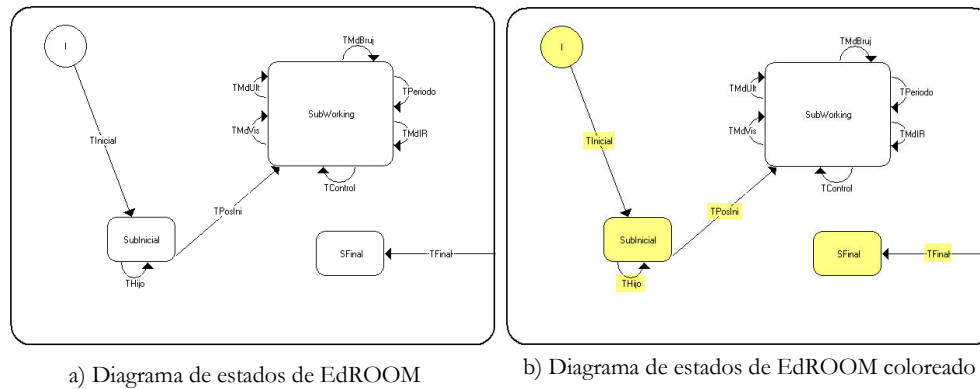


Figura C5-6. Diferencias entre un diagrama de estados en EdROOM y un diagrama de estados coloreado.

En las secciones siguientes se presenta la estructura y funcionalidad de los principales módulos que constituyen la arquitectura de control del robot. Durante la descripción de los módulos se destacarán aquellas decisiones que se han tomado respecto a la estructura y funcionalidad de los módulos con el objeto de incorporar el algoritmo IFAsynMixto a la arquitectura robótica.

- **ModeloRobot: Módulo principal de la arquitectura de control del robot**

El nivel superior de la estructura, representado de acuerdo con la metodología ROOM en la figura C5-7, que en si mismo constituye un módulo de ROOM y por lo tanto su comportamiento está asociado a una tarea, está formado por tres módulos diferentes. Ordenados de acuerdo con el nivel de abstracción alcanzado en el razonamiento, de menor a mayor, se encuentra primero el módulo *ElRobot* cuya misión es solicitar cíclicamente las medidas a los sensores y enviarle las señales de control a los actuadores. En el siguiente nivel de abstracción se encuentra el módulo *Fusion*, responsable de estimar el estado del entorno y la posición del robot a partir de la información sensorial proporcionada por el módulo *ElRobot* y el modelo del mundo y los algoritmos implementados en el módulo *Fusion*. En el nivel superior se encuentra el módulo *ProgControl*, cuya funcionalidad se encuentra directamente asociada con la utilización de la información del entorno y la posición del robot para dirigir al robot a la consecución de los objetivos deseados (seguir un camino, alcanzar un determinado punto del mapa, vagar, etc.).

Esta división del nivel estructural superior de la arquitectura robótica es consecuencia directa de uno de los objetivos principales de diseño del sistema: servir de banco de pruebas para analizar el funcionamiento de algoritmos generales de fusión e integración multisensorial. Se interpreta la búsqueda de generalidad en los algoritmos de fusión como una baja dependencia entre (1) éstos y el

robot sobre el que serán aplicados y, (2) éstos y los objetivos para los que es necesario conocer el estado del robot y del entorno. La división planteada soporta dicha búsqueda de generalidad, ya que permite utilizar los mismos algoritmos para diferentes robots y objetivos, sustituyendo o modificando la funcionalidad de los módulos *ElRobot* y *ProgControl* respectivamente, y definir el comportamiento de los algoritmos de fusión utilizados en el módulo *Fusion*. Aún así, los cambios introducidos en cualquiera de los tres módulos que modifiquen la información intercambiada con otro módulo, harán necesario reformar la parte de la funcionalidad del otro módulo que trabaje sobre dicho tipo de información.

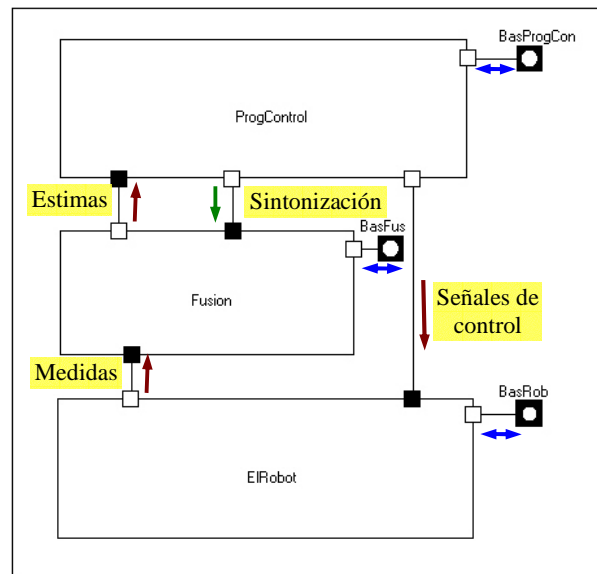


Figura C5-7. Estructura general de la arquitectura de control del robot.

La organización de tareas en este nivel es fundamentalmente jerárquica, a pesar de que el intercambio de información entre dos módulos no se realiza en un único sentido como ocurría en las arquitecturas deliberativas. El punto de partida del lazo de control a este nivel, identificable por medio de las flechas rojas, se encuentra en el módulo *ElRobot* que obtiene las medidas realizadas por el robot, pasa a través del módulo *Fusion* encargado de la percepción y alcanza el módulo *ProgControl* que decide las ordenes de control, de acuerdo con los objetivos deseados, que hay que enviar al robot a través del módulo *ElRobot*. La flecha verde identifica una conexión utilizada para sintonizar los algoritmos de fusión de acuerdo con las decisiones tomadas, por lo que establece una relación de arquitectura híbrida. Los intercambios bidireccionales entre los módulos componentes (*ElRobot*, *Fusion* y *ProgControl*) y el principal, identificables por las flechas azules, que aparecerán también en la definición de todos los módulos del sistema que incluyan otros módulos en la definición de su estructura, han sido establecidos para sincronizar el proceso de inicialización y finalización de todas las tareas del sistema. La funcionalidad del módulo principal se encuentra únicamente relacionada con dicha labor de sincronización.

▪ **ProgControl: Módulo de control, planificación, navegación y supervisión**

El nivel de abstracción superior de la arquitectura de control se encuentra representado por el módulo *ProgControl*, encargado de utilizar la información que el módulo *Fusion* le proporciona sobre el entorno y posición del robot para realizar labores de control, planificación, navegación y supervisión. La funcionalidad asociada a estas labores se encuentra directamente relacionada con los objetivos para los que el robot ha sido diseñado, y puede ser encapsulada dentro del módulo *ProgControl* en diferentes subconjuntos de módulos, niveles de comportamiento, etc. La única restricción existente sobre su estructura es que mantenga la interfaz de puertos externos (denominados *FusionData*, *SintonizarFusion*, *ComandosRobot* y *Basico* en el diagrama estructural de la figura C5-8) y sobre la funcionalidad es que incluya las etapas necesarias para sincronizar el inicio y la finalización de las tareas. La estructura utilizada en los experimentos realizados para analizar el funcionamiento del algoritmo IFAsynMixto dentro de la arquitectura se encuentra recogida en la siguiente figura.

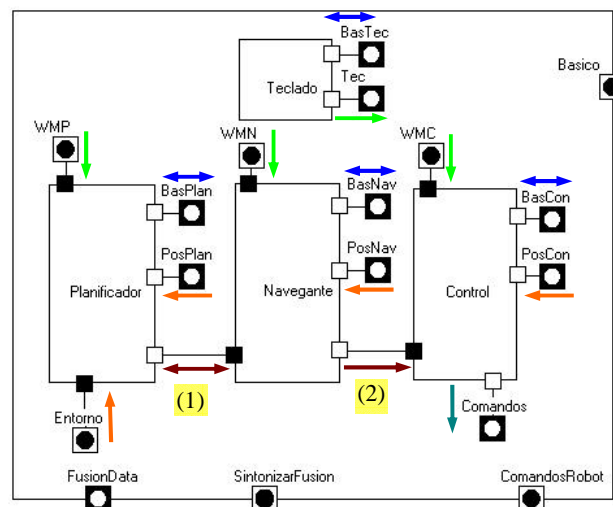


Figura C5-8. Estructura del módulo del control general (*ProgControl*) de la arquitectura robótica.

La funcionalidad asociada a las labores de planificación, navegación, supervisión y control, encapsulada mediante el módulo *ProgControl*, se ha distribuido respectivamente entre los cuatro módulos que lo constituyen: *Planificador*, *Navegante*, *Control* y *Teclado*.

La funcionalidad asociada a la tarea del módulo *ProgControl* se centra en la sincronización de las labores de iniciación y finalización de las tareas de sus módulos constituyentes (para las que se usan los puertos *Basico*, *BasPlan*, *BasNav*, *BasCon* y *BasTec*) y de redistribución de información:

- La información recibida del módulo *Fusion* (a través del puerto *FusionData*) se redistribuye entre el *Planificador*, el *Navegante* y el módulo *Control* a través del puerto *Entorno* (por el

que se reenvía la información del entorno) y a través de los puertos *PosPlan*, *PosNav* y *PosCon* (por los que se reenvía la posición del robot).

- Los comandos generados por el módulo *Control*, que el módulo *ProgControl* recibe a través del puerto *Comandos*, se reenvían al módulo *ElRobot* a través del puerto *ComandosControl*.
- La información recibida del módulo *Teclado* a través del puerto *Tec* es reenviada a los módulos *Planificador*, *Navegante* y *Control* a través de los puertos *WMP*, *WMN* y *WMC*.

El módulo *Teclado* permite al usuario interactuar con el programa a través del teclado para elegir el modo de funcionamiento de los tres módulos restantes. Los modos permitidos son tres: dos manuales, en los que el usuario toma el control del robot, bien a través del teclado o del joystick, y se inhiben las labores de planificación, vigía y control de los módulos *Planificador*, *Navegante* y *Control*; y otro automático, en el que los módulos *Planificador*, *Navegante* y *Control* realizan todas las labores del control de alto nivel del robot. El modo de funcionamiento puede ser modificado en cualquier momento de la ejecución del programa.

Los módulos que constituyen el módulo *ProgControl* se organizan sobre una arquitectura híbrida, con un flujo de información jerárquico entre las conexiones (1) y (2) de los módulos *Planificador*, *Navegante* y *Control* que segmenta el cálculo de las ordenes de control (caminos, trayectoria con restricciones y ordenes de control de los motores respectivamente), y otro flujo distribuido (a través de los puertos *Entorno*, *Pos**, *Tec* y *WM**) sobre la que se soporta un comportamiento reactivo y concurrente de los cuatro módulos. Los diferentes flujos de información se representan en las dos figuras siguientes: en la primera se representa el flujo asociado al comportamiento automático del sistema, y en la segunda, los flujos correspondientes a los comportamientos manuales.

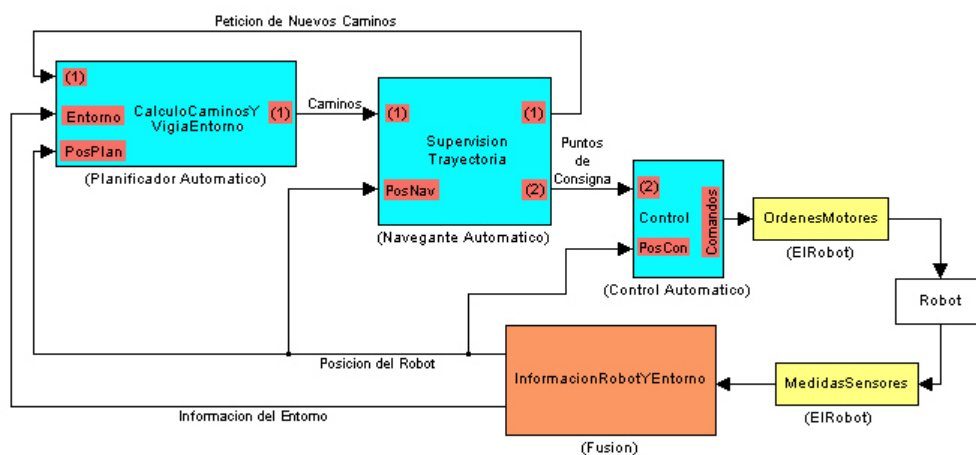


Figura C5-9. Principales flujos de información asociados al comportamiento automático de los módulos componentes del *ProgControl* (representados en azul) y la relación de dichos flujos con la información procedente o enviada al resto de los módulos del primer nivel (*ElRobot* y *Fusión*). Para cada módulo componente del *ProgControl* (azul) se representa también, en su interior y sobre fondo rojo, los puertos de comunicación del módulo *ProgControl* a través de los que intercambia la información.

robot (con la información recibida a través del puerto interno *Entorno*), y utiliza algoritmos de cálculo de trayectoria óptimas, que son enviadas al módulo *Navegante* a través de la conexión (1) de la figura C5-8.

El *Planificador* elegido para las pruebas del algoritmo IFAsynMixto utiliza dos representaciones locales del entorno: un mapa global de todo el entorno en el que se puede mover el robot y un mapa local del entorno próximo al robot en ese instante. El mapa global divide a todo el entorno en celdas iguales en las que se recoge el grado de creencia de ocupación de esa región del espacio y es implementado como una matriz de creencias. El mapa local es representado mediante un “quadtree” ([Jackings1983] y [Latombe1991]) implementado mediante un poliárbol de celdas desiguales, y en cada una de las cuales se establece si la región del espacio correspondiente se considera ocupada, vacía o mixta (ocupada y vacía). El objetivo del robot es trasladarse desde la posición actual hasta una final, elegida por el usuario al inicio de la ejecución del programa. Para conseguirlo el *Planificador* calcula un camino global (sobre el mapa global) que le sirve para orientarse durante la obtención de los caminos locales. Los caminos locales (calculados sobre el mapa local), que son enviados al *Navegante*, esquivan los objetos que han aparecido en el entorno desde la obtención del último camino global. Para el cálculo de ambos caminos se utilizan: (1) un grafo de visibilidad en el que se recogen las distancias de todas las trayectorias posibles entre el robot, el objetivo final del camino, y los vértices y esquinas del mapa correspondiente y (2) el algoritmo de Dijstra para elegir la combinación de caminos más cortos. Las labores de vigía consisten en detectar aquellas celdas del “quadtree” que estando etiquetadas como vacías corresponden a zonas del mapa global cuya creencia ha sido incrementada por encima del valor de corte que diferencia las celdas consideradas vacías de las consideradas ocupadas o mixtas. Para evitar una posible situación de peligro, cuando el *Planificador* detecta un punto de este tipo, le envía al *Navegante* la orden de solicitud de parada del robot y espera un tiempo, calculado en función de la proximidad del nuevo objeto, antes de calcular un nuevo camino local y enviárselo al *Navegante*. Toda esta funcionalidad, realizada cuando el usuario ha solicitado el funcionamiento automático de los módulos *Planificador*, *Navegante* y *Control*, se especifica en el diagrama de estados de la figura C5-11, que forma parte de uno de los niveles de comportamiento desarrollados para el *Planificador*.

Navegante: Módulo de seguimiento y supervisión de trayectorias

La funcionalidad automática del módulo *Navegante* tiene como objeto calcular los puntos de consigna que debe seguir el robot basándose en la trayectoria que le envía el *Planificador* a través de la conexión (1) de la figura C5-8 y en las restricciones de movimiento propias del robot, y comprobar si dichos puntos, que envía al módulo de *Control* a través de la conexión (2) de la misma figura, son seguidos de forma correcta por el robot.

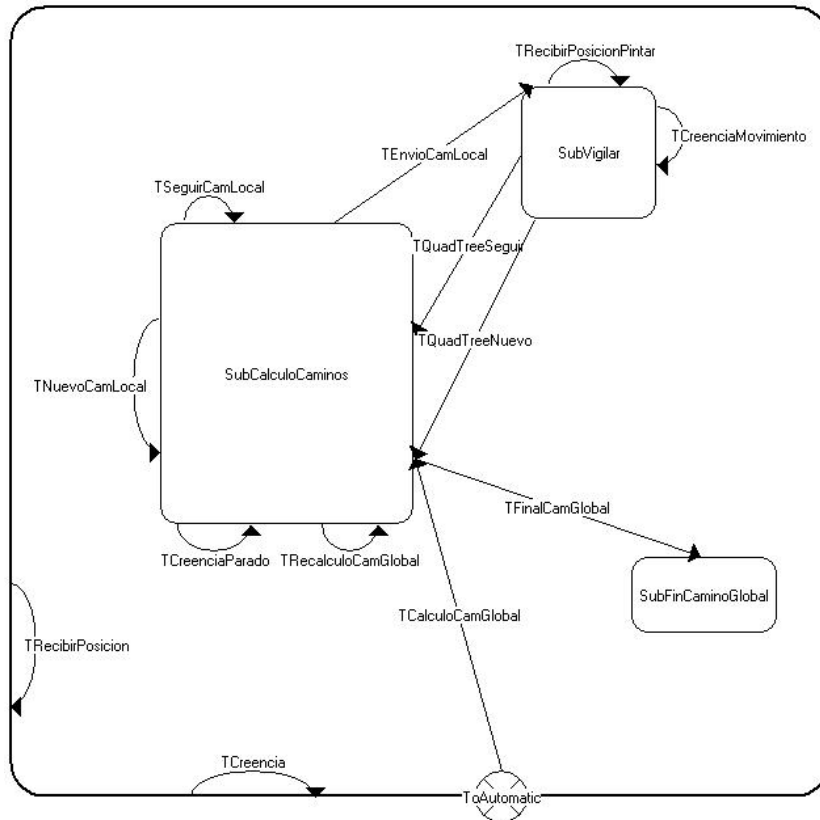


Figura C5-11. Diagrama de estados asociado al funcionamiento automático del módulo *Planificador*

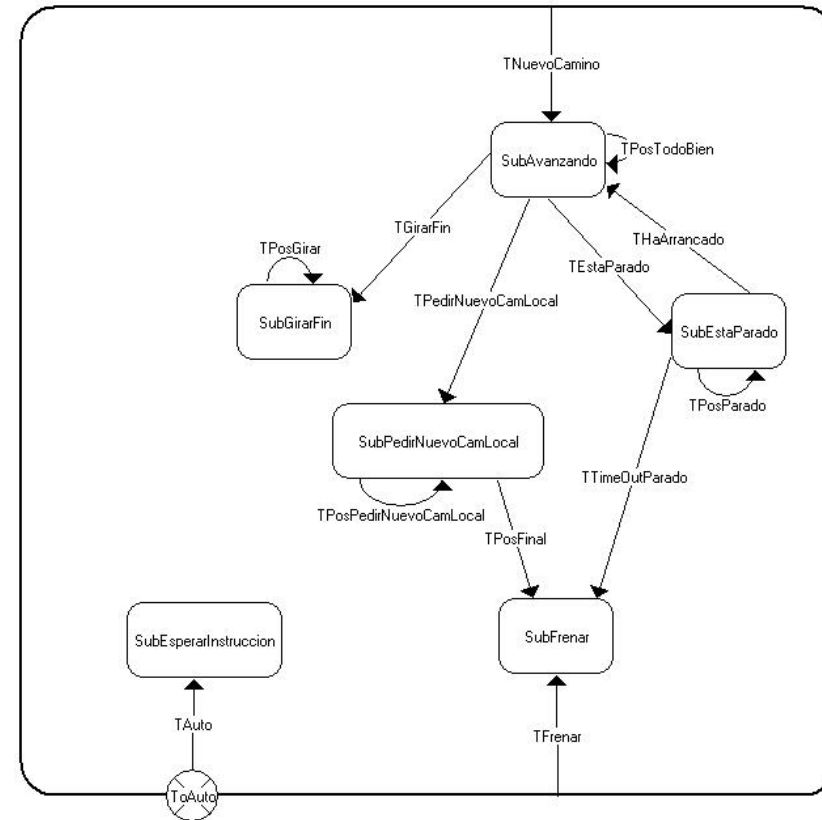


Figura C5-12. Diagrama de estados asociado al funcionamiento automático del módulo *Navegante*

La funcionalidad de la supervisión automática realizada por el *Navegante* se especifica en el diagrama de estados de la figura C5-12, que forma parte de uno de los niveles de comportamiento desarrollados para el *Navegante*. En ella se comprueba, cada vez que el *Navegante* recibe una nueva posición a través del puerto *PosNav*, si el robot se mueve dentro del margen de maniobra del robot o si se encuentra parado durante un tiempo cuando el *Navegante* considera que debería estar siguiendo una trayectoria. En los casos en los que el *Navegante* detecta que se va a llegar al final de la trayectoria local o que ésta no está siendo seguida de forma correcta, solicita al *Planificador* un nuevo camino local. Además, al final de la trayectoria local que sitúa al robot al final del camino global pide al módulo de *Control* que reoriente al robot con el ángulo final deseado.

Control: Módulo de generación de las señales de control de los motores del robot

La funcionalidad automática del módulo *Control* se encarga de calcular la señal de control que hay que aplicar a las ruedas del robot para que desde la posición en la que se encuentra, que recibe periódicamente a través del puerto *PosCon*, se aproxime (y pase) por los puntos de consigna recibidos a través de la conexión (2) de la figura C5-8. Por lo tanto este módulo cierra el lazo de control de la figura C5-13, que es el más corto y próximo al robot de todos los que se establecen a través del funcionamiento automático de los módulos *Planificador*, *Navegante* y *Control*, esquematizados en la figura C5-9.

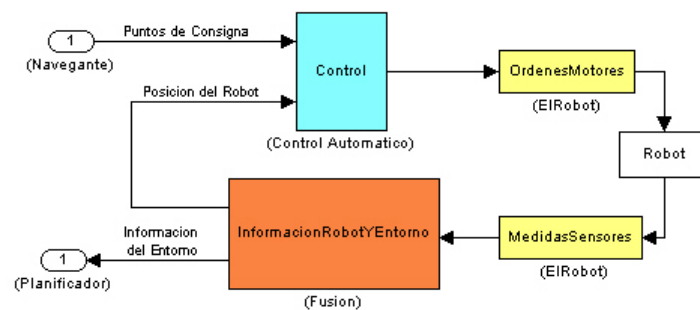


Figura C5-13. Lazo de control en el que interviene el módulo *Control* en modo automático y su relación con el resto de los módulos del sistema que constituyen dicho lazo.

El tipo de ordenes de control obtenidas por este módulo se encuentran fuertemente relacionadas con el sistema de tracción motriz del tipo de robot sobre el que se van a aplicar. Análisis de cinemáticas para distintos robots pueden encontrarse en [Kelly1994] y [Campion1996].

A pesar de esta fuerte dependencia entre la funcionalidad del módulo *Control* y el robot, el módulo *Control* se ha incluido en la estructura del módulo *ProgControl* ya que entendemos que su funcionalidad está directamente relacionada con las labores de control de alto nivel del robot, mientras que los módulos *Fusion* y *ElRobot* se encuentran relacionadas con labores de percepción y de interacción con el robot respectivamente. Soluciones alternativas a la elegida serían colocar este módulo como parte de la estructura del módulo de alto nivel *ModeloRobot* (como si se tratase de un

módulo interfaz entre los módulos *ProgControl* y *ElRobot*), o dentro de la estructura del módulo *ElRobot*. La elección de la posición final de este módulo dentro de la arquitectura de control de robot no determinará la funcionalidad real del sistema, siempre y cuando se establezcan las conexiones necesarias para su correcto funcionamiento. Además, distintos tipos de robots pueden ser controlados con el mismo diagrama de estados utilizando diferentes librerías externas a *EdROOM* para implementar los distintos tipos de algoritmos de control necesarios para cada robot.

La funcionalidad automática del módulo *Control* utilizado en las pruebas del algoritmo IFAsynMixto se especifica en el diagrama de estados de la figura C5-14, que forma parte de uno de los niveles de comportamiento desarrollados para dicho módulo. El algoritmo implementado utiliza un conjunto de reglas que comparan la posición del robot con los puntos de consigna para calcular cíclicamente la velocidad de cada una de las ruedas motrices del robot.

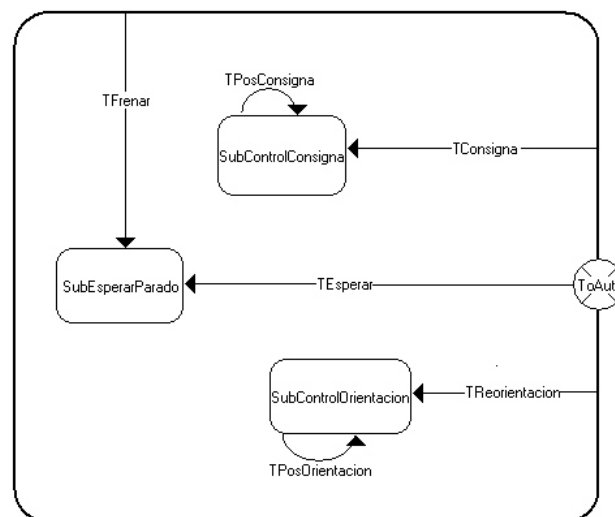


Figura C5-14. Diagrama de estados asociado al funcionamiento automático del módulo *Control*.

- ***ElRobot: Módulo utilizado para interactuar con un robot o simular su comportamiento***

La estructura y el comportamiento del módulo *ElRobot* del nivel superior tienen que estar definidos para poder implementar el sistema. La única restricción existente sobre su estructura es que mantenga la interfaz de puertos externos (denominados *RobotData*, *Comandos* y *Basico* en el diagrama estructural de la figura C5-15) y sobre la funcionalidad es que incluya las etapas necesarias para sincronizar el inicio y la finalización de las tareas (asociadas a los estados y transiciones coloreados en amarillo de la figura C5-16). De esta forma se asegura que el módulo puede ser utilizado con las conexiones establecidas en el nivel superior (*ModeloRobot*), e iniciado y finalizado a la vez que las tareas asociadas al resto de los módulos del sistema. El resto de la estructura y funcionalidad se encuentra directamente relacionado con el robot sobre el que se

realizarán las pruebas, y la representada en las figuras C5-15 y C5-16 es para el robot descrito en el apéndice III de la tesis. Para interactuar o simular la funcionalidad de otros robots habrá que definir la estructura y el comportamiento adecuado por medio de otros módulos del tipo *ElRobot*.

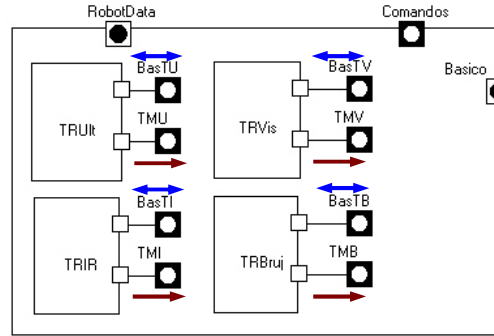


Figura C5-15. Estructura del módulo *ElRobot* utilizado para interactuar con el robot real o simular su comportamiento.

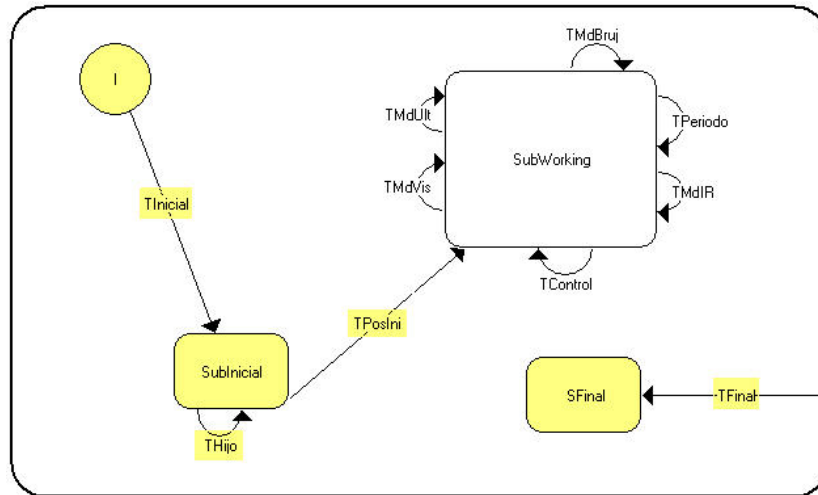


Figura C5-16. Diagrama de estados del módulo *ElRobot* encargado de interactuar con el robot real o simular su comportamiento.

La estructura y el diagrama de estados de las figuras C5-15 y C5-16 sirven para interactuar o simular el comportamiento de cinco de los seis tipos de sensores del robot descrito en el apéndice III (una brújula electrónica, dos encoders ópticos acoplados a las dos ruedas motrices, un cinturón de ultrasonidos, un cinturón de infrarrojos y dos cámaras en B/N para visión estereoscópica) y controlar o simular la respuesta de los dos tipo de actuadores (los motores de corriente continua acoplados a las dos ruedas motrices). La utilización de los sensores de ultrasonidos y el motor paso a paso del sistema triular orientable de ultrasonidos que tiene el robot Nandana no se encuentra contemplado en la estructura y funcionalidad del módulo *ElRobot* utilizado durante los experimentos de esta tesis.

Los cuatro módulos contenidos en la estructura, denominados *TRUlt*, *TRIR*, *TRBruj* y *TRVis*, son responsables de generar cíclicamente las órdenes que provocan que el módulo *ElRobot* solicite las medidas de distancia a objetos al cinturón de ultrasonidos, distancia a objetos al cinturón de infrarrojos, de orientación a la brújula, y de posición de objetos detectados al sistema de visión estereoscópica. Estas órdenes son recibidas por el módulo *ElRobot* como mensajes a través de los puertos internos *TMU*, *TMI*, *TMB* y *TMV*; y son las responsables de disparar las transiciones *TMdUlt*, *TMdIR*, *TMdBruj* y *TMdVis*, durante las que se solicita la medida al dispositivo o sistema correspondiente. La transición *TControl* se dispara cuando se recibe través del puerto externo *Comandos* un mensaje relacionado con el control de los motores acoplados a las ruedas motrices. La transición *TPeriodo* es disparada cíclicamente por un objeto de temporización asociado al módulo *ElRobot*, y durante la misma: (1) se solicitan, recogen y envían a través del puerto externo *RobotData* las medida de los encoders; (2) se aplica la última señal de control recibida durante la transición *TControl* y se envía la señal aplicada a través del puerto externo *RobotData*; y (3) se comprueba si hay una medida disponible de los sensores temporizados por medio de los módulos *TRUlt*, *TRIR*, *TRBruj* y *TRVis*, y si es así, se envía a través del puerto externo *SensorData*. Por último, los puertos *BasTU*, *BasTI*, *BasTB* y *BasTV*, *Basico* son utilizados para intercambiar mensajes para sincronizar la inicialización y finalización de las tareas del sistema.

La funcionalidad asociada a la petición de medidas de los sensores y envío de órdenes a los actuadores se encuentra definida en una librería externa a *EdROOM*. Se han implementado dos librerías externas diferentes para definir el mismo conjunto de clases, de forma que podemos utilizar la misma estructura y diagrama de estados para: (1) simular el funcionamiento del robot (generando la información sensorial en función de un conjunto de modelos de funcionamiento de los sensores y los actuadores), o (2) interactuar con el robot real (solicitando las medidas a los sistemas de información sensorial del robot y enviándole las ordenes a los actuadores).

Por último, es importante destacar que el diferente tratamiento que se le da a los encoders y al resto de los sistemas sensoriales a la hora de pedir y enviar las medidas (para los encoders se pide y se envía durante la transición *TPeriodo*, mientras que para el resto se pide durante las transiciones *TMd** y se envía, si está disponible, durante la transición *TPeriodo*) se encuentra relacionado con el funcionamiento de los sistemas sensoriales y de los algoritmos de integración y fusión implementados en el módulo *Fusion*:

- En los experimentos sobre el robot real, el sistema de visión que determina la posición relativa a las cámaras de los objetos detectados se ejecuta en un PC diferente al PC en el que se ejecuta el programa asociado a la arquitectura de control, y el intercambio de información (petición de información sensorial, envío de la misma, etc.) se realiza a través de una cable cruzado conectado a sus respectivas tarjetas de Ethernet. El programa del PC de visión

realiza medidas y procesa la información continuamente, atendiendo a las peticiones de envío de medidas cada cierto tiempo, por lo que no es conveniente que la tarea asociada al módulo *ElRobot* del programa que implementa la arquitectura de control se quede esperando (en la transición *TMdVis*) a recibir la respuesta del PC de visión. Por lo tanto, en la transición se solicita la información sensorial, y en cada transición *TPeriodo* se comprueba si el PC de control ha recibido respuesta del PC de visión, y si es así, ésta se reenvía al módulo *Fusion* a través del puerto externo *SensorData*.

- El resto de los sensores y actuadores de robot se controlan a través del puerto serie por medio de un conjunto de microcontroladores, utilizando el protocolo de comunicación presentado en el apéndice III, en el que el PC envía una orden a través del puerto serie y se mantiene a la espera hasta recibir la respuesta a la misma. Este mecanismo asegura que tanto las señales de control como la petición y recepción de las medidas se ejecutan de forma ordenada. Este hecho es ventajoso para los sistemas de fusión, ya que las medidas de todos los sensores salvo visión se obtendrán de forma ordenada, pero no es beneficioso para analizar el funcionamiento del algoritmo IFAsynMixto dentro de la arquitectura de control. La existencia del orden en las medidas únicamente es debida al protocolo de comunicaciones establecido con el robot elegido para realizar los experimentos y no es una restricción propia de la arquitectura de control, que puede ser utilizada para robots en los que las medidas se reciban de forma desordenada. Por lo tanto, en el diagrama de estados del comportamiento asignado al módulo *ElRobot* se distinguen dos etapas relacionadas con la brújula y los sensores de ultrasonidos e infrarrojos: en una (transición *TMd**) se realiza la petición de la medida (con la recepción por parte del módulo *ElRobot* del PC de control) y en otra (transición *Tperiodo*) se utiliza un mecanismo de decisión para comprobar si está se encuentra disponible para enviarla al módulo *Fusion* en el que se permite al usuario introducir (simular) la existencia de retardos en la disponibilidad de las medidas.
- Para la petición de medidas de los encoders no se utiliza ni un módulo de temporización ni una transición propia, como ocurre con el resto de los sensores, ya que los modelos del sistema y de medida utilizados en el algoritmo IFAsynMixto, que se exponen detalladamente en el apéndice V, suponen que las medidas de los encoders son realizadas con la misma periodicidad con la que se envían las ordenes de control al robot. Para lograr este comportamiento, durante la transición *Tperiodo* se solicita y recibe la medida de los encoders, ésta es enviada al módulo *Fusion*, y acto seguido se envían a los motores y al módulo *Fusion* la última orden de control recibida por el módulo *ElRobot*. Durante la implementación del comportamiento del módulo *ElRobot* se decidió no simular la existencia de retardos en la disponibilidad de las medidas de los encoders (enviando al módulo *Fusión* las medidas de forma retardada e incluso desordenada) ya que la posibilidad de introducir

retardos en las medidas de los otros sensores es suficiente para analizar el funcionamiento del algoritmo IFAsynMixto.

Hay que tener en cuenta que el comportamiento implementado en el módulo *ElRobot* para introducir retardos en el envío de las medidas desde el módulo *ElRobot* al módulo *Fusion* permite analizar mejor el funcionamiento del comportamiento del algoritmo IFAsynMixto a costa de disminuir el grado de independencia existente entre los dos módulos. De hecho, cuando se estén probando otros algoritmos de integración y fusión multisensorial puede ser necesario modificar el comportamiento asociado al módulo *ElRobot*. Sin embargo, las modificaciones introducidas para poder simular la existencia de retardos, tanto en las pruebas sobre el robot real como sobre la funcionalidad asociada al robot simulado, permiten analizar el funcionamiento del algoritmo IFAsynMixto y de la arquitectura ante la existencia de retardos en la comunicación, y por lo tanto, comprobar si el sistema es adecuado para robots en los que las medidas se reciban de forma retardada y/o desordenada.

▪ ***Fusion: Módulo de integración y fusión multisensorial***

El módulo *Fusion* es el responsable de encapsular e implementar la funcionalidad utilizada para calcular la posición del robot y el estado del entorno a partir de la información proporcionada por los diferentes sensores y/o sistemas sensoriales. Al igual que en los casos de los módulos *ProgControl* y *ElRobot*, la única restricción existente sobre la estructura del módulo *Fusion* es que mantenga la interfaz de puertos externos (denominados *FusionInfo*, *SintonizarFusion*, *Basico* y *RobotData* en el diagrama estructural de la figura C5-17) y sobre la funcionalidad es que incluya los estados necesarios para sincronizar el inicio y la finalización de las tareas (labor para la que se intercambiará mensajes a través de los puertos *Basico*, *BasEnt* y *BasPos*). Además, entre las tareas de inicialización también se incluye el envío de la posición inicial del robot a cada uno de los módulos de fusión (a través de los puertos *PosPos* y *PosEnt*), por si esta información fuese necesaria para la inicialización de los algoritmos implementados en cada uno de los módulos. El resto de la estructura y funcionalidad se encuentra directamente relacionado con los algoritmos de integración y fusión multisensorial utilizados para estimar el estado del robot y de su entorno.

En las pruebas recogidas en esta memoria se utilizan dos algoritmos diferentes para estimar la posición del robot y el estado del entorno. La funcionalidad de cada algoritmo se encapsula en un módulo diferente definido dentro de la estructura del módulo *Fusion*: el algoritmo para estimar la posición del robot se implementa en el módulo *Posicion* mientras que el algoritmo utilizado para estimar el estado del entorno se encapsula en el módulo *Entorno*. Esta división permite analizar el comportamiento de algoritmos diferentes para estimar la posición del robot y el estado del entorno, o algoritmos similares divisibles en dos módulos diferentes. La funcionalidad principal del módulo

Fusion consiste en reenviar la información procedente de los sensores y de los actuadores al módulo de fusión correspondiente (a través de los puertos *SBU*, *SBIR*, *SBV* e *RobotDataPos*), y reenviar cíclicamente la información devuelta por los módulos (a través de los puertos *InfoEntorno* e *InfoPos*) al módulo *ProgControl* (a través del puerto externo *FusionInfo*) y, en el caso que sea necesario, al otro módulo de fusión. Para los algoritmos elegidos durante los experimentos recogidos en esta tesis, la posición del robot tiene que ser enviada al módulo *Entorno* a través del puerto *PosEnt* cíclicamente y a través de los puertos *SBU*, *SBIR* y *SBV* junto con cada medida sensorial asociada al mismo instante de muestreo que es reenviada al módulo *Entorno*.

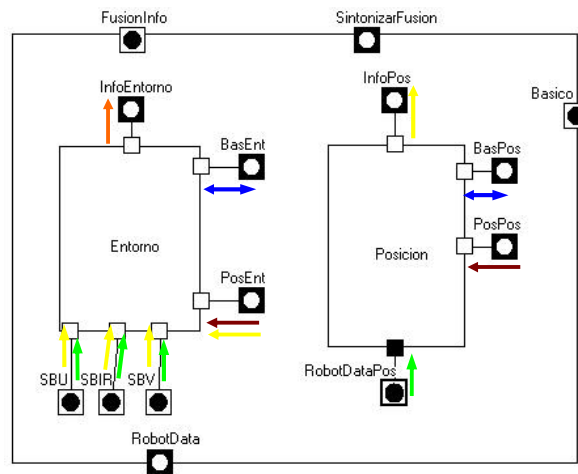


Figura C5-17. Estructura del módulo *Fusion* encargado de procesar la información sensorial con el objeto de estimar la posición del robot y el estado del entorno.

La redistribución de la información procedente del módulo *ElRobot* que se implementa en la funcionalidad del módulo *Fusion* se encuentra ligada al robot utilizado para realizar los experimentos y a los algoritmos implementados en los módulos *Entorno* y *Posicion*. Para el robot utilizado para probar el funcionamiento del algoritmo IFAsynMixto se reenviará al módulo *Entorno* la información procedente del sistema de visión estereoscópica, del cinturón de ultrasonidos, y del cinturón de infrarrojos; mientras que el módulo *Posición* recibirá la información procedente de la brújula, de los encoders, y del cinturón de ultrasonidos, así como las ordenes de control aplicadas a los motores.

En los siguientes apartados se presenta la estructura y funcionalidad asociada a los módulos *Entorno* y *Posicion* utilizados durante las pruebas del algoritmo IFAsynMixto, ambos fundamentados sobre algoritmos de inferencia sobre redes bayesianas.

Entorno: Módulo de fusión multisensorial de las medidas de entorno

El módulo *Entorno* incluido en el módulo *Fusion* encapsula el comportamiento utilizado para estimar el estado del entorno a partir de la información sensorial recibida por los puertos *SBU*,

SBIR y *SBV*. El algoritmo elegido para obtener el estado del entorno determina la parte de la estructura y funcionalidad encapsulada por este módulo que no se encuentra relacionada con las labores de inicialización y finalización de las tareas del sistema y con su interfaz externa de comunicación (representada por los puertos *Entorno*, *Basico*, *Posicion*, *SBU*, *SBIR* y *SBV* de la figura C5-18).

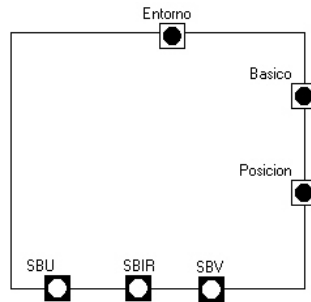


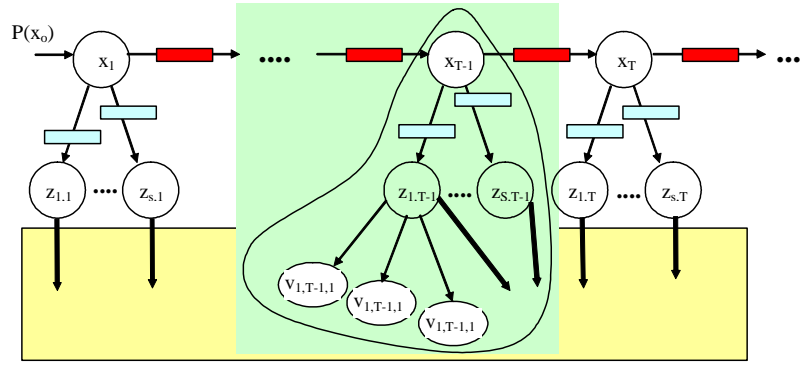
Figura C5-18. Estructura externa del módulo *Mapa* encargado de estimar el estado del entorno.

En las pruebas de este capítulo se utiliza para estimar le estado del entorno el algoritmo multisensorial distribuido de actualización de mapas de creencia basado redes bayesianas descrito en [Lopez-Orozco1998a], [Lopez-Orozco1999] y modificado en [Besada-Portas2002a] para asegurar el correcto funcionamiento del algoritmo cuando existen medidas asíncronas, retardadas y temporalmente desordenadas. El algoritmo almacena la información del estado del entorno en una matriz bidimensional de celdas correspondientes a regiones equiespaciadas del entorno, y utiliza una red bayesiana dinámica y discreta para capturar las relaciones de dependencia entre las variables que describen el estado del entorno y las que son utilizadas para introducir la información proporcionada por los sensores. Este algoritmo ha sido utilizado con éxito en diferentes robots, como son el robot MAR IV, construido en la Universidad de Castilla la Mancha [Lopez-Orozco1999] y los ATVs (All Terrain Vehicles) de la Universidad Carnegie Mellon ([Besada-Portas1999], [Tebi-Ollenum1999]).

En la red utilizada en el algoritmo propuesto, representada en la figura C5-19, el estado de cada celda del mapa \mathbf{x} puede tomar en cada instante t dos posibles valores: vacío u ocupado, que se almacenan en el nodo \mathbf{x}_t bajo la forma de la probabilidad que tiene la celda en encontrarse en cada uno de los estados. El sensor \mathbf{s} puede detectar o no un objeto sobre dicha celda, almacenándose en el nodo $\mathbf{z}_{s,t}$ la probabilidad de detección. El modelo de cada sensor establece la relación probabilística entre las variables \mathbf{x}_t y $\mathbf{z}_{s,t}$, y almacena la probabilidad de detección $P(\text{detectada}|\text{ocupada})$ y de falsa alarma $P(\text{detectada}|\text{vacía})$ del sensor.

La red bayesiana es dinámica ya que su estructura evoluciona a lo largo del tiempo. Las medidas son introducidas en la red como evidencia débil con la probabilidad aportada por el sensor sobre la

detección o no del objeto en dicha celda. Este tipo de evidencia se introduce por medio de la inclusión de los nodos adicionales $\mathbf{v}_{s,t,k}$ (monoestado) que se relacionan con el nodo $\mathbf{z}_{s,t}$ como la probabilidad $P(\mathbf{v}_{s,t,k}|\mathbf{z}_{s,t})$ aportada por el sensor. Además, en los casos en los que los sensores únicamente proporcionen información sobre la presencia de objetos (y no sobre su ausencia) la subred enmarcada en verde en la figura C5-19 no es capaz de contrarrestar la información causada por el funcionamiento erróneo de un sensor o por un objeto que se haya movido y no se encuentre más sobre la celda en la que fue detectado. El problema se puede solucionar utilizando medidas sobre la ausencia de objetos, aunque sobrecarguen el proceso de inferencia, o implementando funciones de olvido por medio de las transiciones temporales entre el estado de la celda entre dos instantes sucesivos ($t-1$ y t). De esta forma, se olvida tanto la información a priori de la celda $P(\mathbf{x}_0)$ como la información proporcionada por los sensores. La velocidad de olvido se elige por medio de la selección de la matriz de transición de estados asociado a la celda $P(\mathbf{x}_t|\mathbf{x}_{t-1})$.



Para cada celda \mathbf{x} del mapa:

- \mathbf{x}_t Estado= $\{\text{Ocupado}, \text{Vacio}\}$ de la celda durante el periodo de tiempo t
- $\mathbf{z}_{s,t}$ Estado de Detección= $\{\text{Detectado}, \neg \text{Detectado}\}$ del sensor s durante el periodo t
- $\mathbf{v}_{s,t,k}$ Variable con un estado para introducir la k -ésima medida del sensor s durante el periodo t como $P(\mathbf{v}_{s,t,k}|\mathbf{z}_{s,t})$
- $P(\mathbf{x}_0)$ Información a priori del estado de la celda
- Modelo del Sensor: $P(\mathbf{z}_{s,t}|\mathbf{x}_t)$
- Modelo del Olvido: $P(\mathbf{x}_t|\mathbf{x}_{t-1})$

Figura C5-19. Red bayesiana dinámica de variables discretas para obtener el estado de la celda \mathbf{x} del mapa.

Para estimar el estado de la celda se pueden utilizar diferentes algoritmos de inferencia exactos, como son los basados en el intercambio de mensajes ([Pearl1988]) o los fundamentados en la construcción del árbol de unión (Lauritzen-Spiegelhalter, Hugin, y Shafer-Shenoy, comparados en [Lepar1998]). El algoritmo utilizado en la versión actual de la arquitectura de control, e implementada por medio de la estructura asociada al módulo *Entorno* de la figura C5-20 y la funcionalidad asociada a sus módulos componentes, se fundamenta en el algoritmo de intercambio de mensajes distribuido propuesto en [Diez1994a]. El módulo *MapaCreencias* almacena la información asociada al estado de ocupación de todas las celdas del mapa, los módulos *NBU*, *NBIR* y *NBV* almacenan la información asociada al estado de detección de cada celda para cada uno de los sensores utilizados para el algoritmo (*NBU* ultrasonidos, *NBIR* infrarrojos y *NBV* visión), y los módulos *SU*, *SIR* y *SV* se utilizan para transformar la información proporcionada por los sensores

en la evidencia débil o probabilidad aportada por el sensor sobre la detección o no del objeto sobre la celda correspondiente.

El algoritmo de inferencia implementada en la versión actual de la arquitectura de control se ha optimizado, reduciendo el número de mensajes intercambiados a aquellos que son necesarios para estimar el estado de las celdas del mapa de creencias. El flujo de dichos mensajes se representa a través de las flechas rojas de la figura C5-20. Cuando un módulo S^* (SU , SIR o SV) recibe una medida de un sensor junto con la posición del robot estimada por el módulo *Posicion* obtiene la celda sobre la que se está midiendo y la probabilidad de detección asociada a dicha medida. Esta información es entonces enviada al módulo NB^* (NBU , $NBIR$ o NBV) al que está conectado. Ante la recepción de este mensaje el módulo NB^* actualiza el estado de detección asociado al sensor para la celda elegida y envía el mensaje correspondiente a esta actualización al módulo *MapaCreencias*, que modificará la probabilidad de ocupación de la celda correspondiente. El nuevo estado de ocupación asociado a cada celda es enviado a través del puerto *Entorno* al módulo del nivel superior *Fusion*.

La probabilidad de ocupación de las celdas también se puede actualizar cíclicamente aplicando una función de olvido en la que se realiza la transición de estado de ocupación de la figura C5-19 sobre un subconjunto de las celdas del mapa próximas al robot. Se ha decidido aplicar el olvido a un subconjunto de celdas en vez de a todo el mapa para evitar olvidar la información de aquellas regiones del mapa que no puedan ser observadas por los sensores desde la posición en la que se encuentra el robot.

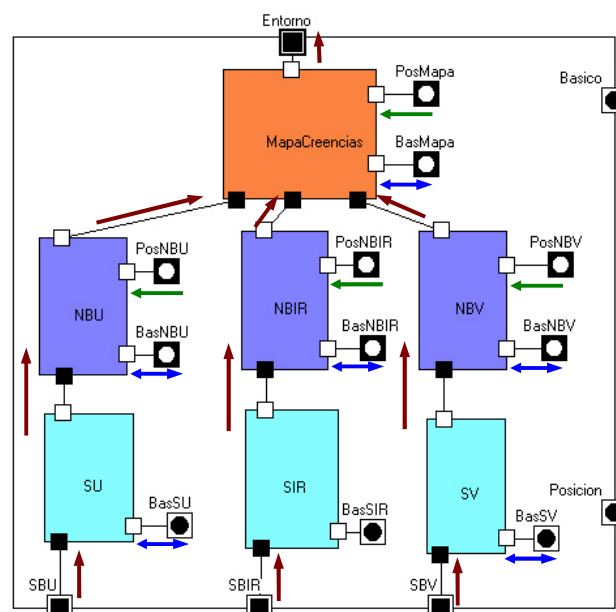


Figura C5-20. Estructura del módulo *Mapa* utilizado para estimar el estado del entorno.

Por último es importante destacar que tanto los módulos S^* como los módulos NB^* recogidos en la estructura del módulo *Entorno* son instancias de sólo dos tipos de módulos diferentes, cuya funcionalidad se implementa con un conjunto de funciones diferentes para cada sensor, elegibles a través de los parámetros de definición de cada tipo de módulo. La estructura del módulo *Entorno* puede ser modificada para fusionar la información procedente de más sensores, a través de la incorporación de más parejas de módulos S^* y NB^* (uno por cada sensor) y de los puertos necesarios para establecer las conexiones fundamentales para el establecimiento del flujo de mensajes de la versión optimizada del algoritmo para la red bayesiana de la figura C5-19.

Posición: Módulo de fusión multisensorial para estimar la posición del robot

El módulo *Posicion* incluido en el módulo *Fusion* encapsula el comportamiento utilizado para estimar la posición del robot a partir de la información sensorial recibida por el puerto *RobotDatosPos*.

En la versión de la arquitectura presentada en [Lopez-Orozco1999] este módulo utiliza el algoritmo de fusión multisensorial presentado en [Lopez-Orozco2000] que basándose en la versión multisensorial del algoritmo de Kalman permite estimar de forma distribuida y robusta la posición del robot a partir de la información proporcionada por un conjunto de sensores acíclicamente (pero no de forma desordenada y/o significativamente retardada).

Para probar el algoritmo IFAsynMixto se sustituye la estructura y funcionalidad de los módulos utilizados en la primera versión de la arquitectura por el módulo cuya estructura y funcionalidad se representan en las figuras C5-21 y C5-22.

En este caso, en vez de dividir la funcionalidad de las diferentes etapas del algoritmo en diferentes módulos y/o diagramas de estado multinivel, se ha optado por utilizar un único módulo (el de la figura C5-21) y diagrama de estados (el de la figura C5-22), junto con el conjunto de clases C++ que encapsulan el comportamiento del algoritmo IFAsynMixto. Ante la llegada de un mensaje procedente del robot a través del puerto externo *RobotData* se dispara la transición *TRobotData*, durante la cual se realiza una etapa de predicción (cuando el mensaje recibido contiene la última señal de control aplicada al robot) o una etapa de validación y asimilación de medida (cuando el mensaje recibido es una medida sensorial). Las posiciones del robot que se pueden ver modificadas por cualquiera de las dos etapas (la del último periodo de muestreo en el caso de recibir la señal de control y la de los periodos de muestreo iguales y posteriores al instante de medida en el caso de recibir una medida) son enviadas al módulo *Fusion* a través del puerto externo *InfoPosicion*. El puerto externo *Basico* es utilizado para la sincronización de las etapas de iniciación y finalización de tareas (cuyos estados y transiciones se representan en amarillo en la figura C5-22) y el puerto *Posición* para recibir la posición inicial del robot, necesaria para la inicialización del algoritmo IFAsynMixto.

Además, ante la posibilidad de que los mensajes de medida referidos a un instante de muestreo lleguen antes que el mensaje de la señal de control asociado al mismo instante, hecho que puede darse ya que el servicio de comunicación de *EdROOM* reordena los mensajes en base a su prioridad, se utilizan unos buffers de almacenamiento de medidas que han llegado antes de tiempo, y en cada transición *TRobotData* se comprueba si las medidas almacenadas en el buffer pueden ser asimiladas (siempre y cuando no resulten ser erróneas).

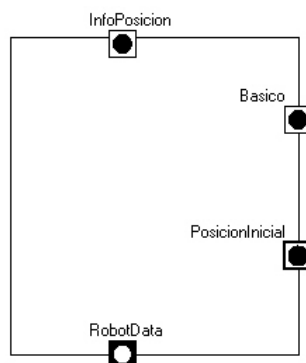


Figura C5-21. Estructura del módulo *Posicion* que implementa la funcionalidad del algoritmo IFAsynMixto.

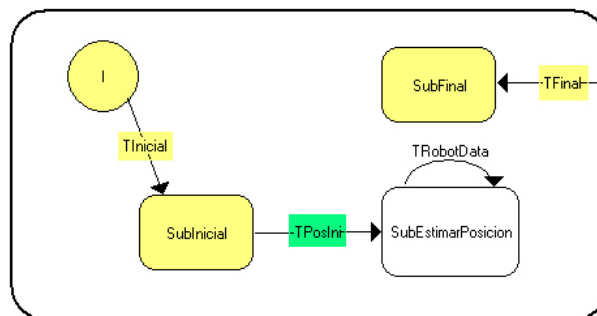


Figura C5-22. Diagrama de estados del módulo *Posicion* que implementa la funcionalidad de IFAsynMixto.

Por último, es importante recordar que para poder utilizar el algoritmo IFAsynMixto para estimar el estado de un sistema es necesario definir un conjunto de funciones (los modelos de transición y de medida), que en el caso de la versión C++ se implementan mediante un conjunto de clases. El robot utilizado está formado por un conjunto de 10 sensores: una brújula electrónica que proporciona información sobre la orientación del robot en el mapa, la pareja de encoders acoplados a las ruedas motrices que proporcionan información sobre el desplazamiento lineal de las ruedas y que serán tratados como un único sensor por el algoritmo de fusión, y los 8 sensores de ultrasonidos que junto con un mapa del entorno sirven para determinar la posición del robot. Las ecuaciones utilizadas para modelar el comportamiento del robot del apéndice III y sus sensores (brújula, encoders, y ultrasonidos del cinturón) se presentan en el apéndice V, junto con un estudio de las características de la información aportada por cada sensor y de las modificaciones que hay

que realizar en el algoritmo IFAsynMixto para asegurar la correcta asimilación de toda la información proporcionada por cada uno de los sensores.

C5.2. Resultados obtenidos por el algoritmo IFAsynMixto en la arquitectura robótica propuesta

La incorporación del algoritmo IFAsynMixto a la arquitectura de control para robots permite realizar un análisis del funcionamiento de dicho algoritmo cuando forma parte de un programa de control de un sistema real. La doble dependencia existente entre la posición estimada por el módulo *Posicion de Fusion* y la señal de control calculada automáticamente de forma conjunta por los módulos constituyentes del módulo *ProgControl* permite estudiar la robustez del algoritmo cuando es utilizado para estimar la posición de un sistema, en este caso un robot, controlado en lazo cerrado.

La arquitectura de control elegida sirve para realizar experimentos sobre diferentes robots, aunque los módulos *ElRobot*, *Entorno* y *Posicion* han sido adaptados para trabajar sobre el robot descrito en el apéndice III.

En este apartado se presentan los resultados obtenidos por el algoritmo IFAsynMixto cuando forma parte de la arquitectura de control del robot, y cuando los módulos de control de alto nivel, encapsulados en el módulo *ProgControl*, funcionan automáticamente, cerrando el lazo de control para que el robot actúe de acuerdo con el objetivo global del sistema (trasladarse de forma segura desde una posición inicial hasta una final) y con los cambios de estado del entorno (mapa de ocupación) y del robot (su posición).

Los resultados recogidos en este apartado han sido obtenidos mediante simulación (utilizando en *ElRobot* la librería externa de simulación) e interactuando de forma real sobre el robot (utilizando la librería externa de control real):

- Los resultados obtenidos mediante simulación permiten analizar el funcionamiento de la arquitectura de control junto con el algoritmo IFAsynMixto cuando se utilizan las mismas ecuaciones para simular el comportamiento de los sensores y los actuadores del robot, y para estimar su posición dentro del filtro. Este hecho nos permite comprobar el funcionamiento del algoritmo en una situación óptima, al coincidir los modelos del sistema y de los sensores, con sus comportamientos simulados.

Además, la simulación favorece la utilización simultánea de mayor número de sensores, al permitir que se consideren despreciables los tiempos necesarios para interactuar con el robot y sus sensores. Es decir, no restringe el análisis del funcionamiento de los algoritmos a las limitaciones inherentes al sistema sobre el que serán probados en los experimentos reales.

Los resultados simulados de este capítulo son diferentes a los resultados obtenidos en las simulaciones realizadas en el capítulo anterior, no sólo por la utilización de unas ecuaciones de modelo de movimiento y de medida diferentes, sino principalmente porque en este capítulo se realiza un control en lazo cerrado del sistema ([Kuo1995], [Ogata1998]) sobre el que se analiza el comportamiento del algoritmo.

- Los resultados obtenidos en las pruebas reales permiten estudiar el funcionamiento de la arquitectura de control junto con el algoritmo IFAsynMixto sobre un sistema real. En este caso, los resultados obtenidos dependen de la completitud y corrección de los modelos de movimiento y de medida utilizados para estimar la posición del robot. Por lo tanto, en las pruebas reales no sólo se analiza la bondad del algoritmo, sino que además se comprueba la exactitud de los modelos propuestos en el apéndice V para el robot del apéndice III.

En las dos secciones siguientes se presentan algunos de los resultados obtenidos por el algoritmo IFAsynMixto durante los experimentos realizados con la arquitectura de control utilizando la versión real y simulada del código de interacción (simulación) correspondiente al robot del apéndice III.

Durante el análisis de todos los experimentos (y en particular de las gráficas en las que se representa el mapa final junto con la trayectoria estimada) es importante tener en cuenta que el módulo *Control* considera que el robot ha alcanzado su posición final cuando el robot se encuentra en las “proximidades” de la misma, entendiéndose durante los experimentos realizados que el robot ha alcanzado su posición final cuando se encuentra situado a menos de 40 cm de la misma.

C5.2.1. Resultados simulados

En esta sección se presentan los resultados obtenidos por el algoritmo IFAsynMixto bajo un conjunto de simulaciones, todos ellos caracterizados por la utilización de:

- Las mismas posiciones de partida del robot y objetivo final, representadas en la gráfica (a) de la figura C5-23.
- El mismo mapa probabilístico de ocupación inicial en los módulos *Entorno* (y *Planificador*), representado en las gráficas (a) y (b) de la figura C5-23 como un plano y en tres dimensiones (los puntos (x,y) corresponden a las celdas del plano y el eje z de la gráfica (b) a la probabilidad de ocupación de la celda correspondiente).
- El mismo mapa de objetos en el modelo del sistema ultrasonidos + mapa del algoritmo IFAsynMixto para estimar la posición del robot. El mapa utilizado se representa en la gráfica (a) de la figura C5-24. Sus objetos corresponden al mapa probabilístico de ocupación inicial.

C5.2. Resultados obtenidos por el algoritmo IFAsynMixto en la arquitectura robótica propuesta

- Las mismas matrices de covarianza para modelar los errores del modelo de movimiento (\mathbf{Q}_t) y de medida de los sensores ($\mathbf{R}_{s,t}$). Estas matrices, especificadas en el apéndice V y cuyos valores variables se especifican en la siguiente tabla para los experimentos simulados, son usadas en el algoritmo IFAsynMixto para estimar la posición del robot y en el módulo *ElRobot* para simular el movimiento del robot y las medidas de los sensores.

Covarianzas de los Modelos de			
Movimiento	Brújula	Encoders	Ultrasonidos
$\text{var}_{\Delta l}(v_L, v_R) = (0.02 \cdot a)^2$ $\text{var}_{\Delta \theta}(v_L, v_R) = (2\pi \cdot a / 180)^2$ $\text{cov}_{\Delta \Delta \theta}(v_L, v_R) = 0$ $a = \begin{cases} 0.1 & \text{si } (v_L = 0) \wedge (v_R = 0) \\ 1 & \text{c.c.} \end{cases}$	$\text{var}_{\text{brújula}} = (\pi / 180)^2$	$\text{var}_{N_L} = (0.01)^2$ $\text{var}_{N_R} = (0.01)^2$	$\text{var}_{\text{sonar}} = (0.01)^2$

Tabla C5-1. Valores de las matrices de covarianza de los modelos de movimiento y de medida utilizados por los sensores que proporcionan información sobre la posición del robot durante los experimentos simulados.

Los valores de $\text{var}_{\Delta \theta}$ y $\text{var}_{\text{brújula}}$ se muestran en rad^2 , mientras que los de $\text{var}_{\Delta l}$, var_{N_L} , var_{N_R} y $\text{var}_{\text{sonar}}$ se presentan en m^2 .

Al utilizar en todos los experimentos el mismo mapa de probabilidad inicial y posiciones de partida y objetivo final, el planificador global implementado en el módulo *Planificador* calcula inicialmente para todos los experimentos el mismo camino global (representado en negro en la gráfica (a) de la figura C5-23 y en las gráficas (a) y (b) de la figura C5-24). Las posiciones inicial y final elegidas hacen que en los ejemplos en los que el robot se mueva siguiendo el camino global inicial la orientación del robot se encuentre cerca de la discontinuidad angular existente entre 0 y 2π . Este hecho permitirá comprobar el correcto funcionamiento de la versión del algoritmo IFAsynMixto recogida en el apéndice V.

La diferencia entre los experimentos recogidos en esta sección se encuentra:

- En el mapa de objetos utilizados en el módulo *ElRobot* para simular el funcionamiento de los sensores de ultrasonidos, infrarrojos y sistema de visión. Los dos mapas de objetos utilizados se representan en las gráficas de la figura C5-24: el mapa de objetos de la izquierda corresponde a los objetos disponibles en el mapa probabilístico de ocupación de la figura C5-23 mientras que el mapa de objetos de la derecha incluye un objeto cilíndrico adicional, en medio de la trayectoria del robot. La utilización de este segundo mapa permitirá: (1) comprobar la reactividad de la arquitectura de control ante los cambios imprevistos del entorno, y ver como una vez que se detecta la existencia del objeto planificará una trayectoria alternativa al camino global inicial; y (2) comprobar como el sistema de ultrasonidos + mapa de objetos utilizado en el algoritmo IFAsynMixto es capaz de desechar las medidas de los objetos de ultrasonidos que no se encuentran en el mapa de objetos

utilizados en el algoritmo IFAsynMixto (que para todos los experimentos es el de la izquierda de la figura C5-24).

- En los sensores utilizados durante los experimentos: para comprobar el funcionamiento del algoritmo IFAsynMixto y la influencia de los distintos sensores en los valores de posición estimados, los sensores se irán incorporando de forma sucesiva a lo largo de los experimentos analizados en esta sección.

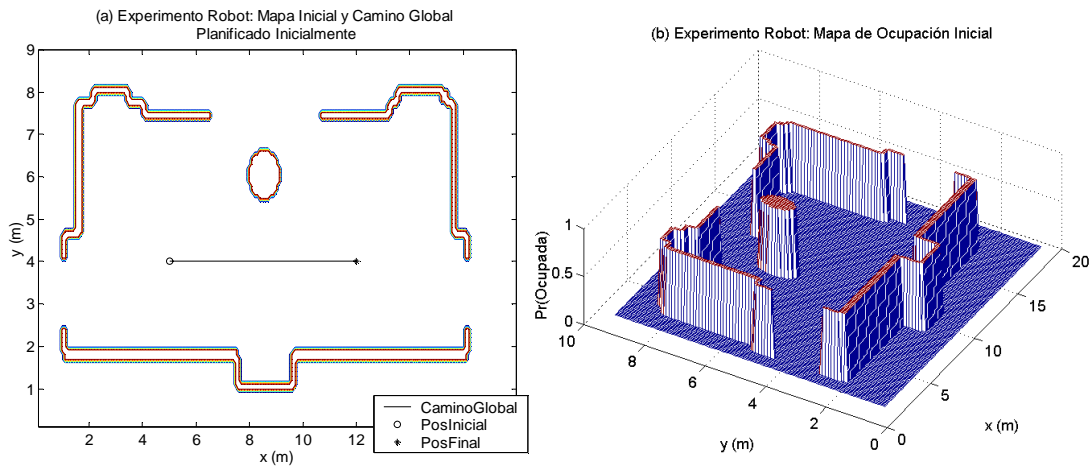


Figura C5-23. Mapa probabilístico de ocupación inicial y punto de partida y objetivo final del robot para todas las simulaciones. En la gráfica (a) se representa el mapa probabilístico de celdas como un plano, el punto de partida del robot (PosInicial), el solicitado como posición objetivo final (PosFinal), y el camino global propuesto inicialmente por el módulo *Planificador*. En la gráfica (b) se representa el mapa probabilístico de celdas en tres dimensiones: los valores del eje z corresponden a la probabilidad de ocupación de cada celda (x,y) del mapa.

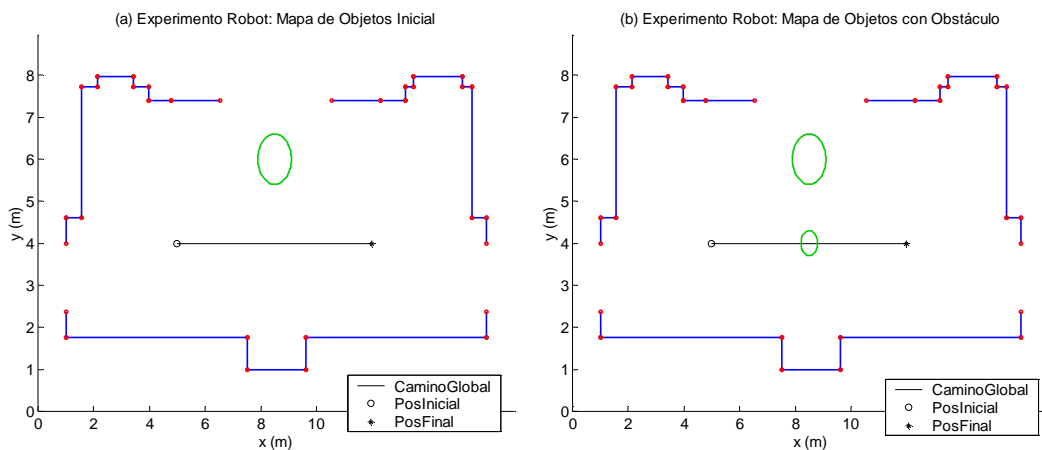


Figura C5-24. Mapas de Objetos utilizados durante los experimentos simulados junto con las posiciones inicial y final del robot, y el camino global planificado inicialmente. En azul se representan las paredes, en rojo las esquinas y los rincones y en verde los objetos cilíndricos. El sistema ultrasonidos + mapa de objetos incorporado en el algoritmo IFAsynMixto utiliza siempre el mapa de la gráfica (a) mientras que el sistema mapa de objetos utilizado para simular las medidas de los ultrasonidos, infrarrojos y sistema de visión puede ser el de la gráfica (a) o el de la gráfica (b).

C5.2. Resultados obtenidos por el algoritmo IFAsynMixto en la arquitectura robótica propuesta

En la siguiente tabla resumen se representa para cada experimento el mapa de objetos utilizado en la simulación de los sensores de ultrasonidos, infrarrojos y sistema de visión, y se indican los sensores utilizados en cada experimento, junto con la periodicidad de las medidas y los retardos mínimos introducidos por el módulo *ElRobot* en la comunicación de las medidas al sistema de fusión multisensorial. El periodo básico de la señal de control (y por lo tanto de los encoders, cuando se estén utilizando) es de 0.2 segundos. Además, todas las medidas recibidas por el algoritmo IFAsynMixto pasarán un test de validación chi-cuadrado.

Experimento	Mapa Sim	Encoders		Brújula		Ultrasonidos		Infrarrojos		Visión	
		Per	Ret	Per	Ret	Per	Ret	Per	Ret	Per	Ret
Sin sensor	--	--	--	--	--	--	--	--	--	--	--
Encoders	--	0.2	--	--	--	--	--	--	--	--	--
Brújula+Encoders	--	0.2	--	2.0	1.0	--	--	--	--	--	--
Todos (sin obstáculo)	(a)	0.2	--	2.0	1.0	2.0	0.0 a 0.6	1.4	0.2 a 0.4	5.0	3.0
Todos (con obstáculo)	(b)	0.2	--	2.0	1.0	2.0	0.0 a 0.6	1.4	0.2 a 0.4	5.0	3.0

Tabla C5-2. Características diferenciables de los experimentos simulados. En la primera columna se recoge el nombre identificativo del experimento, en la segunda el mapa de objetos de la figura C5-24 utilizado para simular las medidas de los tres últimos sensores de la tabla, y en las cinco restantes la información asociada al periodo (per) de muestreo (en seg) y los retardos (ret) de comunicación mínimo asociado de cada sensor (en seg).

▪ Experimento simulado sin sensores

En el primer experimento realizado se desactiva el funcionamiento de todos los sensores, por lo que el estado del entorno no se verá modificado y la posición del robot se estimará con el algoritmo IFAsynMixto utilizando únicamente el modelo de movimiento del robot. La falta de medidas elimina la realimentación real del lazo cerrado, y a que no existe ninguna diferencia alguna entre la posición esperada por el módulo Control (en función de las ordenes que generó) y la estimada por el algoritmo IFAsynMixto. Los resultados obtenidos se presentan en las gráficas de la figura C5-25, en los que se puede comprobar el aumento de la varianza debido a la falta de medidas y la influencia de la falta del lazo cerrado de control en la posición real del robot.

Para conseguir que el robot, que se encuentra inicialmente orientado en la dirección del camino global, siga dicho camino, el módulo *Control* calcula señales de control iguales para ambas ruedas. El robot se mueve siguiendo estas señales con un determinado error, que el algoritmo IFAsynMixto recoge en la creciente varianza de los estados (representada en la gráfica (b) de la figura C5-25) y no en su valor estimado. La trayectoria estimada (representada en las gráficas (a) y (c) de la misma figura respectivamente como los tres estados frente al tiempo y el estado x frente a y) según el modelo de movimiento cuando las señales de control de ambas ruedas son iguales es una línea recta. Por lo tanto, como en el módulo *Control* se utiliza la trayectoria estimada y éste considera que el robot se mueve en línea recta (a pesar del creciente desconocimiento sobre su posición real) seguirá generando señales de control iguales para ambas ruedas. Y así sucesivamente. Esto se debe

a que la falta de medidas elimina la realimentación del sistema, y por lo tanto, el sistema se controla sin tener en cuenta la posición real en la que se encuentra el robot. Para poner de manifiesto la importancia de la falta del lazo cerrado, en este experimento también se representa la posición real del robot (en la gráfica (c) de la figura C5-25), calculada por el módulo *ElRobot* de acuerdo con las ecuaciones de movimiento y su matriz de covarianza.

En la gráfica de la varianza se observa un aumento significativamente mayor para el estado y que para el estado x . Esto es causado por las ecuaciones no lineales de modelo de movimiento y la matriz de covarianza elegidas (y presentadas en el apéndice V). Éstas hacen que cuando el robot se mueva paralelo al eje x , los errores de desplazamiento lineal y angular, y del estado angular se acumulen más significativamente en el estado y que en el x .

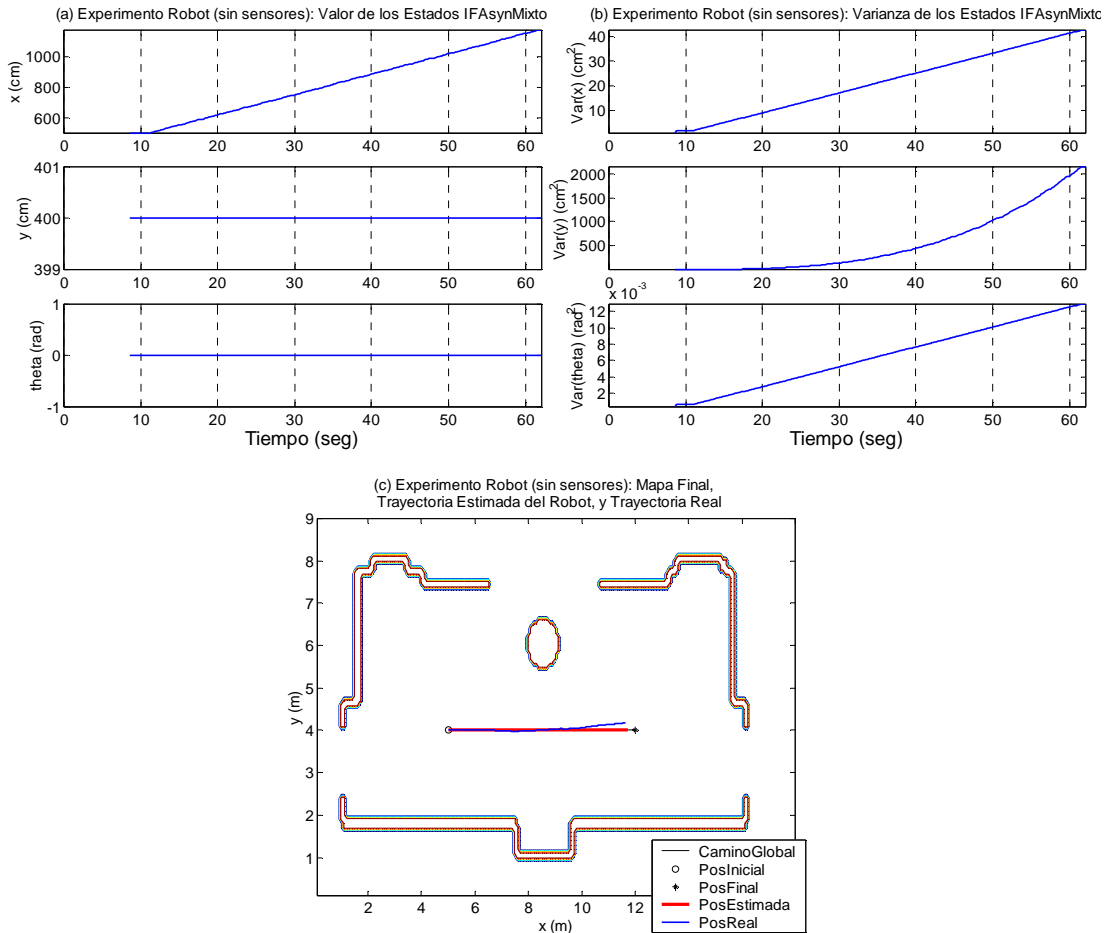


Figura C5-25. Resultados del experimento sin sensores. En la gráfica (a) se representa el valor estimado de los estados (x , y , θ), en la gráfica (b) la varianza asociada a dichos estados, y en la gráfica (c) el mapa de probabilidad de ocupación final (que es igual al inicial) junto con la trayectoria estimada y real del robot.

▪ **Experimento simulado con medidas de los Encoders**

En el segundo experimento se activa el funcionamiento de los encoders acoplados a las ruedas motrices del robot, que proporcionan información sobre el desplazamiento realizado por cada una de las ruedas. Estos valores se encuentran linealmente relacionados con los desplazamientos lineales y angulares del robot (componentes cuarta y quinta del vector de estados del modelo utilizado). El periodo de muestreo de los encoders es de 0.2 segundos, y la lectura de las medidas de dichos sensores coincide con los instantes de aplicación de la señal de control. El módulo *ElRobot* no introduce retardos intencionados en la comunicación de las medidas, aunque es posible que el servicio de comunicación de *EdROOM* si lo haga.

El estado del entorno no se modifica, ya que el sensor activado no proporciona información sobre el mismo. La existencia de las medidas proporcionadas por un sensor establecerá la realimentación del lazo de control, hecho que se pone de manifiesto en este experimento. Los resultados obtenidos se presentan en las gráficas de la figura C5-26, y en los mismos también se puede observar el correcto funcionamiento del algoritmo en torno a la discontinuidad $0-2\pi$.

En la gráfica (a) se representa, para cada instante de aplicación de la señal de control, que ocurre con las medidas proporcionadas por los encoders. La nomenclatura elegida para el eje de ordenadas es similar a la utilizada en los experimentos simulados descritos en el capítulo anterior: *V* representa la existencia de una medida válida, *NV* representa la existencia de una medida no válida, *MT* representa la existencia de una medida que ha llegado fuera de la ventana de tiempo del algoritmo IFAsynMixto, *NM* representa la ausencia de medida, *NC* representa la existencia de una medida que no coincide con un instante de aplicación de señal de control, *T&V* representa la existencia de una medida válida que llega retardada al algoritmo IFAsynMixto y *T&NV* representa la existencia de una medida no válida que llega retardada al algoritmo IFAsynMixto.

En la gráfica (a) se observa que la mayor parte de las medidas, todas salvo las cuatro primeras, han llegado al sistema en el instante de muestreo correspondiente (ya que son *V* y *NV*), y de las mismas casi todas han superado el test de validación con éxito. El resto de las medida (las cuatro primeras en este ejemplo) han sufrido una suerte diferente: la primera ha sido desechada por llegar fuera de la ventana de tiempo, la segunda no coincide con el instante de muestreo, la tercera es válida aunque ha llegado tarde debido a los retardos introducidos por el servicio de comunicación de *EdROOM* y la cuarta no se ha producido o no ha sido recibida por el algoritmo IFAsynMixto.

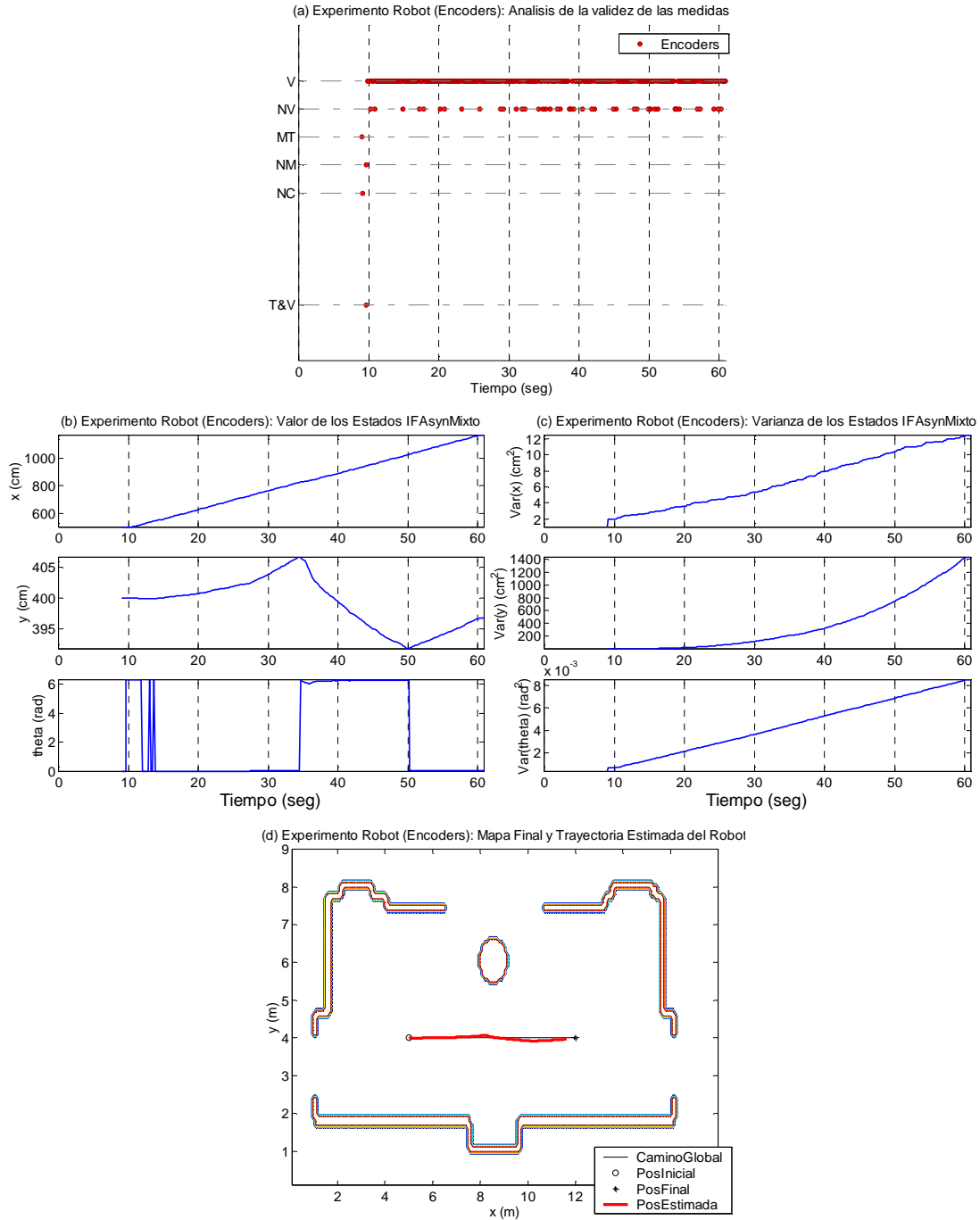


Figura C5-26. Resultados del experimento con encoders. En la gráfica (a) se representa para cada periodo de muestreo de la señal de control la suerte sufrida por la medida (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, NC no coincide con un instante de muestreo, T&V tarde y válida). En (b) se muestra el valor estimado de los estados (x , y , θ), en (c) su varianza, y en (d) el mapa de probabilidad de ocupación final (que es igual al inicial) junto con la trayectoria estimada del robot.

La existencia de tantas medidas válidas, que son asimiladas por el algoritmo IFAsynMixto, modifica los resultados obtenidos en la etapa de predicción del filtro e influye en el cálculo de la señal de control. Cada medida asimilada corrige los valores del vector de estado y reduce el

desconocimiento que se tiene sobre el mismo. Las correcciones en la posición hacen que el algoritmo sea capaz de estimar la falta de rectitud en la trayectoria del robot (observable en las gráficas (b) y (d) de la figura C5-26) y en consecuencia, que el módulo *Control* calcule señales de control diferentes para cada rueda con el objeto de corregir las desviaciones (en desplazamiento y orientación) estimadas respecto al camino global. El movimiento oscilante en torno a la trayectoria observable en la gráfica (c) es causado por el algoritmo de control implementado, basado en un sencillo conjunto de reglas.

La asimilación de las medidas disminuye el desconocimiento que se tiene sobre la posición del robot, almacenado en la matriz de covarianza del estado. Por lo tanto, las curvas de varianza asociada a los estados (x, y, θ) , representadas en la gráfica (c) de la figura C5-26, toman valores menores que las curvas de varianza asociadas al experimento sin medidas de los sensores, representadas en la gráfica (b) de la figura C5-25. Aún así, la varianza asociada al estado y toma valores significativamente mayores a la varianza asociada al estado x , ya que el robot se mueve próximo a la paralela al eje x y a que la información de los encoders permite reducir los incrementos de varianza asociados al desplazamiento lineal más significativamente que los asociados al desplazamiento angular.

Por último es importante destacar que los saltos observables en el valor del estado θ de la gráfica (b) de la figura C5-26 corresponden a cambios de valores en torno a la discontinuidad $0-2\pi$ (en torno a la que se mueve la orientación del robot cuando este se traslada siguiendo un camino paralelo al eje x). Éstos son debidos a la utilización de la función de normalización de la versión adaptada del algoritmo IFAsynMixto (presentada en el apéndice V), aplicable en la etapa de predicción y en la asimilación (de las medidas de los encoders), y consiguen que los valores angulares se encuentren siempre en el rango permitido.

▪ **Experimento simulado con medidas de los Encoders y de la Brújula**

En el tercer experimento se activa el funcionamiento de un sensor adicional: la brújula, que proporciona directamente información sobre la orientación del robot (tercera componente del vector de estados del modelo de movimiento utilizado). A pesar de la linealidad existente entre la medida del sensor y el estado, las medidas proporcionadas por este sensor tienen que sufrir un tratamiento especial (explicado en el apéndice V) debido a la discontinuidad angular $0-2\pi$. Además, la información asociada a las mismas debe de ser recalculada en el caso en el que se reciban medidas retardadas procedentes de otros sensores. Por lo tanto, *los resultados de esta sección, presentados en las gráficas de la figura C5-27, nos permitirán comprobar el correcto funcionamiento de la versión del algoritmo IFAsynMixto adaptada para asegurar el tratamiento especial de las medidas angulares y su correcto funcionamiento en torno a la discontinuidad $0-2\pi$.*

Durante este experimento el estado del entorno no se modifica, ya que los sensores activados (encoders y brújula) no proporcionan información sobre el mismo.

La configuración del funcionamiento de los encoders explicado en la sección anterior no se modifica. La brújula toma medidas cada 2 segundos y el módulo *ElRobot* introduce retardos intencionados en la comunicación de las medidas de 1 segundo, que pueden ser incrementados ocasionalmente por el servicio de comunicación de *EdROOM*. *Debido a la utilización del algoritmo IFAsynMixto, la existencia de retardos en las medidas de la brújula no influye, salvo por la influencia del desorden en la etapa de validación y en el cálculo de la señal de control, en los resultados obtenidos.* La diferencia entre los resultados obtenidos en esta sección y en la anterior se fundamenta en la capacidad del algoritmo para asimilar la información retardada de la brújula, ya que si ésta asimilación no fuese posible, el algoritmo sólo podría disponer de la información de los encoders, al igual que en la sección anterior, aunque en este caso, si hubiese medidas de la brújula. Los resultados obtenidos, y analizados a continuación, se representan en la figura C5-27.

En la gráfica (a) de la figura C5-27 se representa, para cada instante de aplicación de la señal de control, que ocurre con la brújula y los encoders:

- El comportamiento de los encoders es similar al de la sección anterior. Hay una medida disponible para cada instante de aplicación de la señal de control, salvo para los dos primeros en los que la medida no se ha realizado (*NM*) o se ha realizado y ha llegado al algoritmo fuera de la ventana de tiempo (*MT*). Para el resto de los instantes de control hay una medida disponible que llega al algoritmo IFAsynMixto en el instante de muestreo en el que tienen que ser asimilada. La mayoría de dichas medidas son válidas (*V*), y un conjunto pequeño no puede ser asimilada al no superar el test de validación (*NV*).
- La brújula únicamente proporciona medidas cada dos segundos, por lo que en la mayoría de los instantes de aplicación de la señal de control no hay una medida disponible (*NM*). En los instantes en los que hay una medida y como consecuencia al retardo introducido por el módulo *ElRobot* en la comunicación de las medidas, éstas llegan retardadas al algoritmo IFAsynMixto. En este caso, todas ellas superan el test de validación (*T&V*).

En las gráficas (b) y (d) de la figura C5-27 se representa la trayectoria estimada por el algoritmo utilizando todas las medidas válidas (*V* y *T&V*). Al igual que en el ejemplo de la sección anterior, estas medidas permiten corregir los valores del vector de estado, reducir la varianza asociada a los mismos y estimar la falta de rectitud en la trayectoria del robot. Por lo tanto, hacen que el módulo *Control* genere señales para corregir las desviaciones estimadas respecto al camino global.

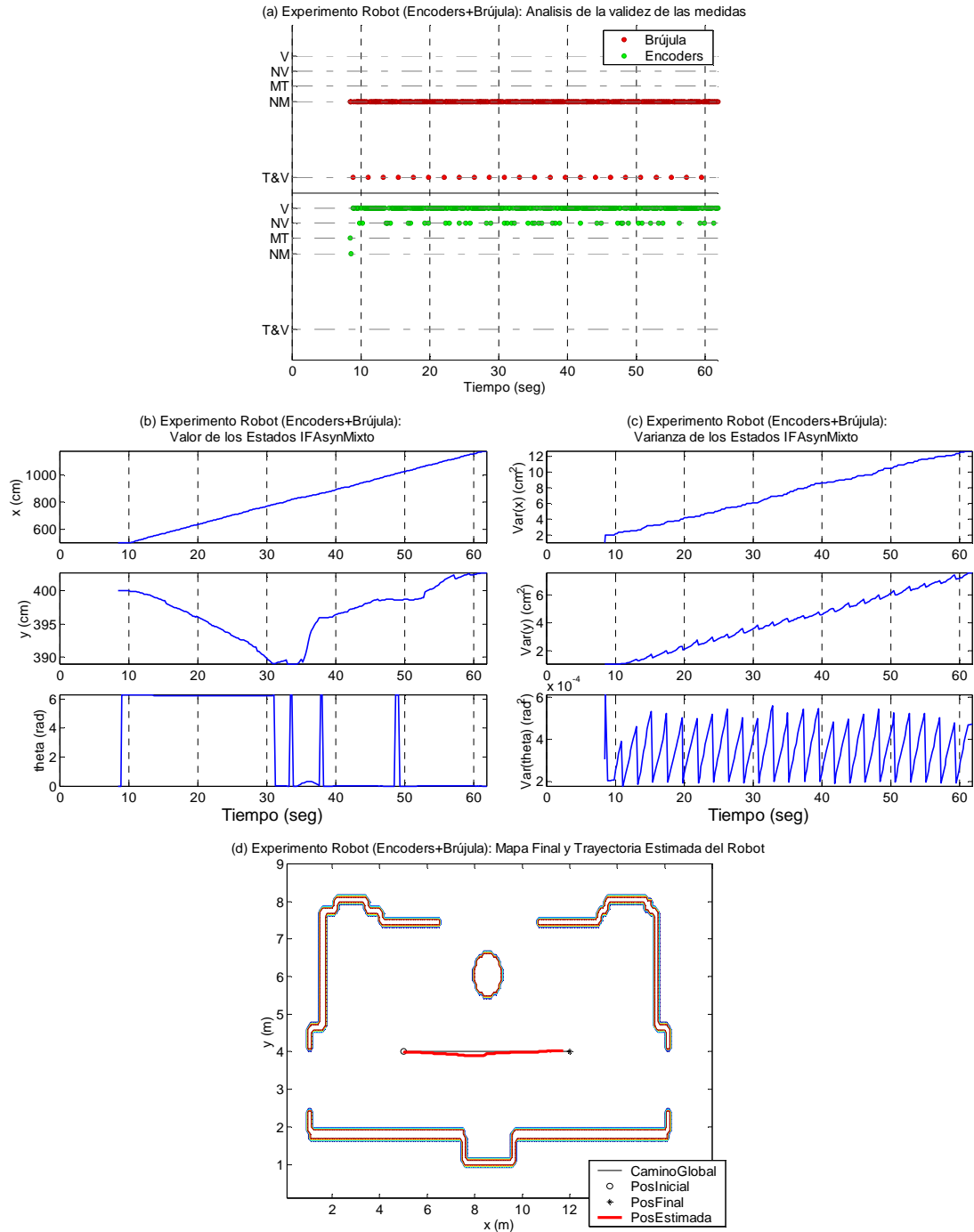


Figura C5-27. Resultados del experimento con encoders y brújula. En la gráfica (a) se representa para cada periodo de control la suerte sufrida por las medidas (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, T&V tarde y válida). En (b) el valor estimado de los estados (x , y , θ), en (c) la varianza asociada a dichos estados, y en (d) el mapa de probabilidad de ocupación final (que es igual al inicial) junto con la trayectoria estimada del robot.

La asimilación de las medidas proporcionadas por la brújula, que llegan retardadas al algoritmo, disminuye cada 2 segundos el desconocimiento que se tiene sobre la orientación del robot, y por lo tanto, el valor reduce cíclicamente el valor de la varianza asociada a dicho estado (representado en la gráfica (c) de la figura C5-27). Aún más, la reducción del error de orientación evita el

crecimiento significativo de la varianza asociada al estado y que se observaba en los dos experimentos anteriores como consecuencia de la traslación del robot en torno al camino global inicial. Es importante recordar que las reducciones de la varianza de los estado y y θ se deben a la capacidad del algoritmo IFAsynMixto para asimilar la información retardada (en este caso relativa a las medidas de la brújula).

Los valores de la varianza del estado x se mantienen similares a los valores observados durante el experimento en el que únicamente se utilizaban los encoders (gráfica (c) de la figura C5-26), ya que según la dirección de la trayectoria realizada, el error en la x se debe principalmente al error en el desplazamiento lineal del robot y no al del desplazamiento angular.

Por último, recordar que los saltos observables en el valor del estado θ de la gráfica (b) de la figura C5-27 corresponden a cambios de valores en torno a la discontinuidad $0-2\pi$ (en torno a la que se mueve la orientación del robot cuando éste se traslada siguiendo un camino paralelo al eje x). Estos cambios bruscos, que hacen que los valores se encuentren próximos a los dos extremos la discontinuidad, muestran como la forma de calcular la información asociada a las mismas previa a su asimilación, presentada en el apéndice V, funciona correctamente.

▪ ***Experimento simulado con medidas de todos los sensores sin obstáculo en el mapa de objetos utilizado en la simulación***

En el cuarto experimento se activa el funcionamiento de los ultrasonidos, cuyas medidas son utilizadas simultáneamente para actualizar el estado del entorno (dentro del módulo *Entorno* del sistema de fusión) y estimar la posición del robot con el algoritmo IFAsynMixto (implementado en el módulo *Posicion*) haciendo uso del mapa de objetos inicial (sin obstáculo, representado en la gráfica (a) de la figura C5-24). Como el mapa de entorno va a ser modificado por las medidas del cinturón de ultrasonidos, decidimos activar el resto de los sensores de entorno (infrarrojos y el sistema de visión estereoscópica) para reducir el número de experimentos que hay que realizar para comprobar el funcionamiento de la arquitectura cuando todos los sensores se encuentran activados. Al mismo tiempo, se analiza la respuesta del algoritmo IFAsynMixto cuando se añaden a las medidas de la brújula y los encoders, las proporcionadas por los sensores de ultrasonidos. Para simular las medidas de los tres tipos de sensores de entorno el módulo *ElRobot* utiliza un mapa de objetos del entorno, que en el experimento propuesto en esta sección coincide con el mapa de objetos utilizado en el algoritmo IFAsynMixto, y cuyos objetos se encuentran recogidos en el mapa de probabilidad de ocupación inicial. *De esta forma en este experimento se puede observar como se actualiza el mapa del entorno del robot y la influencia de la asimilación de medidas procedentes del sistema ultrasonidos + mapa en los resultados del algoritmo IFAsynMixto.*

El modo de funcionamiento de los encoders y la brújula no se modifica respecto al experimento anterior. Los sensores que se activan en este experimento toman medidas cíclicamente y el módulo *ElRobot* introduce retardos en la comunicación:

- El sistema de visión toma medidas cada 5 segundos y cuando detecta objetos en las mismas, envía la posición de los objetos detectados (relativa a la posición de las cámaras) al módulo *Fusion* con un retardo de 3 segundos. El retardo introducido por el módulo *ElRobot* sirve para simular el tiempo de procesamiento de las medidas y el retardo de comunicación existente entre el PC de visión y el PC de control. Debido a los retardos que aparecen entre la realización de la medida y la recepción del resultado por el módulo *Fusion*, éste buscará el valor estimado de la posición del robot asociada al instante de medida, y lo enviará junto con el valor de la medida al módulo *Entorno*.
- Cada grupo de cuatro sensores del cinturón de infrarrojos realiza una medida cada 1.4 segundos. Si tras la realización de las medidas sucesivas de dos grupos, separadas entre si por 0.2 segundos, se ha detectado algún objeto, el módulo *ElRobot* envía las medidas asociadas a las detecciones con un retardo de comunicación adicional de 0.4 o 0.2 segundos (el primer retardo asociado a las medidas del primer grupo y el segundo a las del segundo). Al ser una medida de entorno retardada, el módulo *Fusión* buscará la posición estimada para el instante de medida y la reenviará junto con la información recibida al módulo *Entorno*.
- Cada grupo de dos sensores del cinturón de ultrasonidos realiza una medida cada 2.0 segundos. Si tras la realización de las medidas sucesivas de cuatro grupos, separadas entre si por 0.2 segundos, se ha detectado algún objeto, el módulo *ElRobot* envía las medidas asociadas a las detecciones con un retardo de comunicación adicional de 0.6 a 0.0 segundos (el primer retardo asociado a las medidas del primer grupo y el segundo a las del último). El algoritmo *Fusion* reenvía las medidas recibidas tanto al módulo *Entorno* (para las que buscará las posiciones estimadas para cada instante de medida, posiciones que reenviará al módulo *Entorno* junto con la información recibida) como al módulo *Posicion* (donde el retardo de las medidas es tratado conveniente por los mecanismos establecidos en el algoritmo IFAsynMixto).

El mapa de probabilidad de ocupación final (representado en las gráficas (a) y (b) de la figura C5-28) *no difiere significativamente del mapa inicial* (representado en las gráficas (a) y (b) de la figura C5-23) *ya que los objetos detectados por los sensores de entorno ya se encuentran recogidos en el mapa de probabilidad de ocupación inicial*. Las diferencias, más fácilmente observables en la gráfica (b), se encuentran en los picos que aparecen sobre la columna y algunas paredes, causados por el incremento de probabilidad de ocupación, debido a que los sensores confirman la existencia de los objetos recogidos en el mapa de probabilidad inicial.

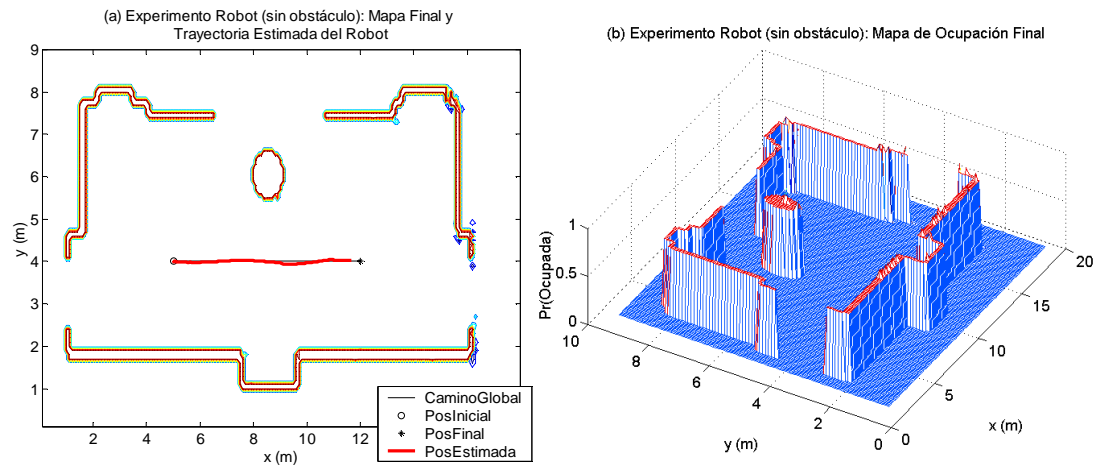


Figura C5-28. Mapa probabilístico de ocupación final y trayectoria estimada del robot para el experimento con todos los sensores funcionando y con un obstáculo en el mapa de objetos usado en la simulación. En la gráfica (a) se representa el mapa probabilística final de celdas como un plano, el punto de partida del robot (PosInicial), el solicitado como posición objetivo final (PosFinal), el camino global propuesto inicialmente por el módulo *Planificador*, y la trayectoria estimada por el algoritmo IFAsynMixto. En la gráfica (b) se presenta el mapa probabilístico final en tres dimensiones: los valores del eje z representan la probabilidad de ocupación de cada celda (x,y) del mapa.

Los cambios producidos en el mapa durante el desplazamiento del robot no modifican el entorno próximo al robot de una forma significativa que haga que el módulo *Planificador*, responsable de vigilar los cambios de estado en el entorno, considere necesario que el robot modifique el camino global planificado inicialmente. Por lo tanto, el módulo *Control* generará señales que le permitan seguir el mismo camino que en los experimentos anteriores y corregir las desviaciones de la trayectoria estimada (representada esta última en la gráfica (a) de la figura C5-28).

En la figura C5-29 se representa que sucede con los encoders y la brújula para cada instante de aplicación de la señal de control. En la misma se puede observar que:

- Los encoders mantienen un comportamiento similar al observado en los experimentos anteriormente expuestos. Proporcionan medidas para todos los instantes de muestreo (salvo uno inicial, NM), y la mayor parte de las mismas puede ser procesada por el algoritmo IFAsynMixto (salvo la que llega fuera de la ventana de tiempo MT o ha sido realizada en un instante de tiempo no asignable a un instante de aplicación de señal de control NC). El resto de las medidas llegan de forma ordenada al algoritmo IFAsynMixto, y la mayor parte de las mismas superan el test de validación (V).
- La brújula proporciona medidas al algoritmo cada 2 segundos que llegan al algoritmo de forma temporalmente retardadas ($T\&^*$), y la mayor parte de las mismas supera el test de validación chi-cuadrado ($T\&V$).

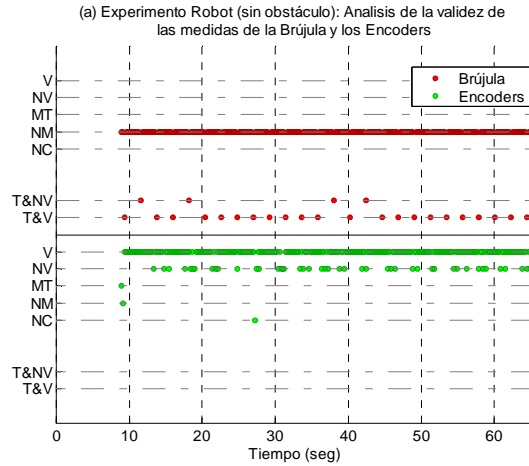


Figura C5-29. Análisis de las medidas proporcionadas por los encoders y la brújula en el experimento con todos los sensores funcionando y con un mapa de objetos para la simulación sin obstáculos. Para los instantes de muestreo de la señal de control se muestra la suerte sufrida por las medidas (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, T&NV tarde y no válida, T&V tarde y válida).

Los sensores de ultrasonidos proporcionan al algoritmo IFAsynMixto medidas (retardadas) sobre la distancia existente entre el robot y los objetos detectados. Cuando se dispone de un mapa inicial de objetos, estas medidas pueden ser utilizadas, para determinar la posición del robot en función de los objetos del mapa utilizando un algoritmo que asocie las medidas a los objetos del mapa. El modelo utilizado, descrito en el apéndice V, considera la existencia de cuatro tipos de objetos: paredes, columnas, rincones y esquinas. Cuando no se puede establecer una asociación entre una medida y un objeto del mapa, ésta es desechada. Como el resto de las medidas utilizadas por el algoritmo IFAsynMixto, las medidas asociadas a objetos tienen que pasar el test de validación chi-cuadrado antes de ser asimiladas.

Por lo tanto, el proceso de validación de las medidas de ultrasonidos está formado por dos etapas, la primera realizada en el proceso de asociación de las medidas a objetos del mapa y la segunda al test chi-cuadrado. Para analizar lo sucedido con las medidas de ultrasonidos, cuyo comportamiento se representa en las gráficas (a) y (b) de la figura C5-30, se utiliza también la información almacenada por el algoritmo IFAsynMixto referente al tipo de objeto asignado a las medidas válidas y las causas por las que las medidas no válidas han sido rechazadas, junto con la posición de los ultrasonidos en el cinturón del robot. Toda esta información, representada en las gráficas de la figura C5-30, permite realizar un análisis exhaustivo del proceso de asimilación y rechazo de las medidas de ultrasonidos. En las gráficas (c) y (d) de la figura C5-30 se utilizan, para todos los instantes de la señal de control, las siguientes etiquetas para representar el comportamiento del algoritmo de asociación: NMide si no hay medida o si la hay y no es asimilable a ningún instante de muestro, NRecon si hay medida pero no es asociable a ningún objeto del mapa, (Pared) si la medida es asociable a una pared, (Colum) si la medida es asociable a una superficie cilíndrica, y (Esq/Rin) si la medida es asociable a una esquina o un rincón. En la gráfica (e) de la misma figura se representa la posición de los sensores del robot orientado 0° respecto al eje x.

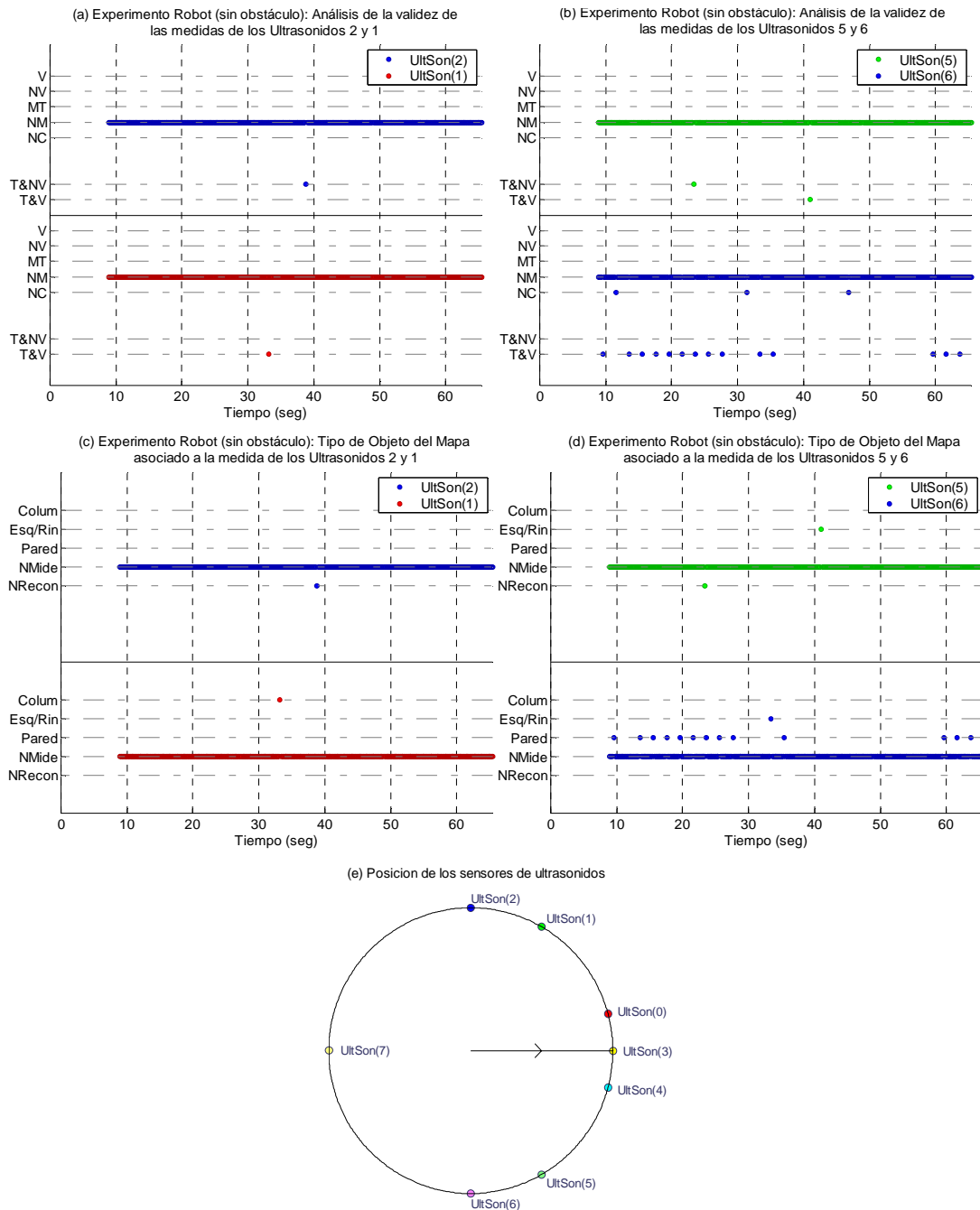


Figura C5-30. Análisis, para el experimento de mapa de objetos de simulación sin obstáculo, de la validez de las medidas de los ultrasonidos, y del tipo de objeto que el sistema de ultrasonidos + mapa asocia a cada medida de los sensores de ultrasonidos. En las gráficas (a) y (b), para cada instante de muestreo de la señal de control, se muestra la suerte sufrida por las medidas (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, T&NV tarde y no válida, T&V tarde y válida). En las gráficas (c) y (d) se representa si no hay una medida procesable (NMide), si a pesar de tener una medida procesable el sistema ultrasonidos + mapa no es capaz de asociarla a ningún objeto del mapa (NRecon), o si la medida existe y es asociada a una Pared (Pared), Columna (Colum), o esquina o rincón (Esq/Rin). Los sensores de ultrasonidos que no han proporcionado ninguna medida (UltSon(0), UltSon(3), UltSon(4), UltSon(7)) no se encuentran representados en las gráficas. En la gráfica (e) se representa la distribución de los sensores sobre el robot.

Del estudio de la información recogida en las cinco gráficas de la figura C5-30, junto con el valor de la posición estimada del robot (representado en la gráfica (a) de la figura C5-31), se concluye que:

- Los ultrasonidos, a pesar de tomar medidas cíclicamente cada 2 segundos, únicamente detectan objetos en alguna de las mediciones realizadas. Los instantes de aplicación de señal de control, etiquetados como *NM* en las gráficas (a) y (b) de la figura C5-30, corresponden a instantes en los que no se ha medido o detectado ningún objeto. Casi todas las medidas correspondientes a la detección de objetos, realizadas por los sensores de ultrasonidos 1, 2, 5 y 6, pueden ser asignadas a un instante de muestreo de la señal de control (salvo las del sensor 6 identificadas como *NC*) y llegan al algoritmo IFAsynMixto temporalmente retardadas ($T \& *$). El proceso de validación, que en este caso está formado por dos etapas, hace que algunas medidas puedan ser utilizadas para determinar la posición del robot ($T \& V$) y otras sean rechazadas ($T \& NV$).
- La única medida realizada por el sensor de ultrasonidos 2 en la que se detecta un objeto no es válida ($T \& NV$) ya que el algoritmo de asociación no es capaz de asociarla a ningún objeto del mapa (*NRecon*). La primera medida del sensor 5 en la que se detecta un objeto presenta el mismo comportamiento.
- La única medida realizada por el sensor de ultrasonidos 1 en la que se detecta un objeto es válida ($T \& V$) ya que el algoritmo de asociación del sistema ultrasonidos + mapa la relaciona a la columna (*Colum*) del mapa de objetos original (gráfica (a) de la figura C5-24) y supera con éxito el test de validación chi-cuadrado. La segunda medida del sensor 5 en la que se detecta un objeto (esquina, de acuerdo con la asignación del algoritmo de asociación) presenta un comportamiento similar.
- Todas las medidas del sensor de ultrasonidos 6 en las que se detectan objetos, salvo aquellas que no son asociables a un instante de muestreo de la señal de control (*NC*), llegan al algoritmo IFAsynMixto retardadas y son válidas ($T \& V$), ya que el algoritmo de asociación del sistema ultrasonidos + mapa las asocia a una pared (*Pared*) o una esquina o rincón (*Esq/Rin*). Teniendo en cuenta la posición del sensor y la orientación del robot durante la trayectoria, las paredes a las que se asocia el objeto son las paralelas al eje x que se representan en la parte inferior del mapa de objetos original (gráfica (a) de la figura C5-24).

Los resultados de la asimilación de las medidas válidas procedentes de la información ordenada y desordenada de la brújula, encoders y ultrasonidos se representan en las gráficas de la figura C5-28. La trayectoria estimada, representada en la gráfica (a), es parecida a la de los ejemplos de las secciones anteriores. La *varianza*, representada en la gráfica (b), presenta un comportamiento diferente debido a la asimilación de las medidas de ultrasonidos y al rechazo de parte de las medidas de la brújula:

- La varianza del estado x decrece bruscamente en los instantes en los que se asimila la información asociada a las columnas y a los rincones o esquinas. La asimilación de información asociada a las paredes no reduce la varianza de este estado ya que el robot se mueve paralelo al eje x y, por lo tanto, la distancia aportada por el sensor permite determinar la posición y del robot y no la x .
- La varianza del estado y decrece cada vez que se asimila una medida de ultrasonidos asociada a cualquier tipo de objeto: paredes, columna, y esquina o rincón.
- La varianza del estado θ presenta un comportamiento similar a la de los experimentos anteriores, con reducciones bruscas cada vez que se asimila una medida de la brújula. Los 3 picos más elevados de este ejemplo se deben a la falta de la asimilación de las medidas rechazadas por el test chi-cuadrado. Además, las pequeñas reducciones que se observan sobre los dientes de sierra de la varianza de este estado son debidos a que las medidas de ultrasonidos asimiladas también proporcionan información sobre la orientación del robot.

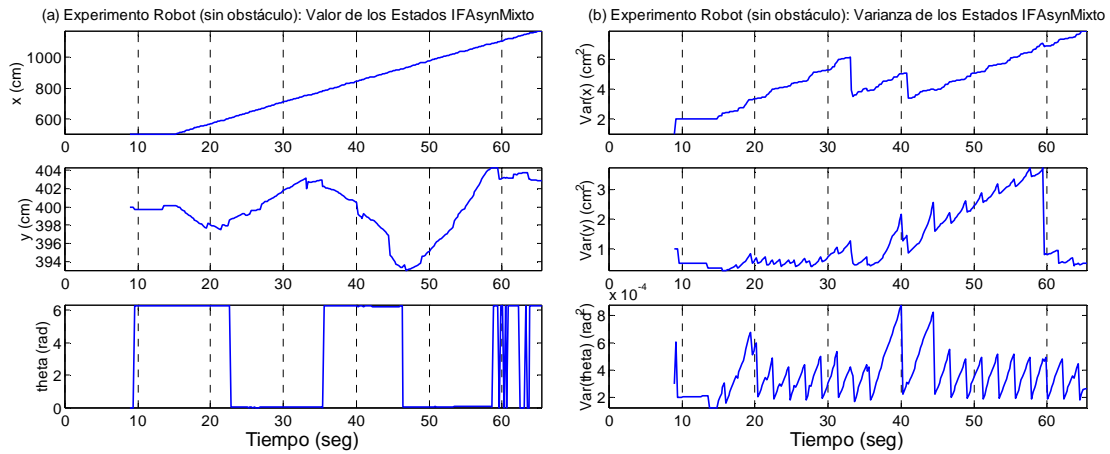


Figura C5-31. Valor estimado de los estados (x , y , θ) y su varianza para el experimento con todos los sensores funcionando y con un mapa de objetos para la simulación sin obstáculo.

Por lo tanto, la información de los ultrasonidos resulta de utilidad para estimar el estado del sistema ya que permite reducir la varianza de los estados. En este ejemplo se observa que tanto la no linealidad del sensor ultrasonidos + mapa, con modelos diferentes para cada tipo de objeto, como el retardo de comunicación variable de las medidas de los sensores de ultrasonidos y la brújula son manejados de forma correcta por el algoritmo IFAsynMixto. Aún más, se comprueba el funcionamiento de la versión adaptada del algoritmo (recogida en el apéndice V) cuando trabaja con 10 sensores diferentes (1 brújula, 1 pareja de encoders y 8 ultrasonidos), y tiene que recalcular la información sensorial asociada a 9 de los mismos (1 brújula + 8 ultrasonidos) en el caso de existir medidas desordenadas.

▪ **Experimento simulado con medidas de todos los sensores con obstáculo en el mapa de objetos utilizado por el módulo ElRobot**

En el último experimento simulado se utilizan dos mapas de objetos diferentes: el primero (presentado en la gráfica (a) de la figura C5-24) se utiliza para generar el mapa de ocupación probabilístico inicial y para la asociación de medidas de ultrasonidos a objetos de mapa en el algoritmo IFAsynMixto, y el segundo (gráfica (b) de la misma figura) para generar en el módulo ElRobot las medidas de los sensores de ultrasonidos, infrarrojos y sistema de visión. La única diferencia existente entre ellos es la presencia en el segundo mapa de una columna adicional, situada en el camino global que el Planificador calculará al inicio de la ejecución del programa. La existencia de dicho objeto permitirá comprobar simultáneamente la reactividad de la arquitectura de control frente a cambios del entorno y si el algoritmo de asociación del sistema de ultrasonidos + mapa del algoritmo IFAsynMixto es capaz de rechazar las medidas de objetos detectados que no existen en su mapa inicial.

El modo de funcionamiento de todos los sensores activados (encoders, brújula, ultrasonidos, infrarrojos y sistema de visión) no se modifica respecto al experimento anterior.

En el mapa de probabilidad de ocupación final (representado en las gráficas (a) y (b) de la figura C5-32) se observa la existencia de un nuevo objeto (u objetos) en el mapa, en las celdas correspondientes a la posición del obstáculo en el mapa de objetos. La probabilidad de dicho objeto no es tan elevada como el de los objetos existentes en el mapa de ocupación inicial (gráfica (b) de la figura C5-23) debido a que la probabilidad inicial de la zona es mucho menor y a que los sensores no han estado detectando el objeto continuamente. Además, en el mapa de ocupación final también aparecen picos de probabilidad de ocupación sobre la columna y algunas paredes, causados por la detección de dichos objetos con los sensores de entorno.

Los cambios producidos en el mapa durante el desplazamiento del robot hacen que el Planificador, en su labor de vigía de los cambios en el entorno, recalcule un nuevo camino alternativo para esquivar las nuevas zonas con una probabilidad de ocupación significativamente superior a la inicial. Este camino alternativo se calcula como un conjunto de caminos locales, que se van modificando según el robot va observando los cambios en su entorno próximo. El robot sigue el camino planificado inicialmente durante una parte de la trayectoria ya que el objeto que se está detectando se encuentra fuera del primer camino local, y por lo tanto, el Planificador no ve necesario modificar la trayectoria del robot hasta que éste se encuentre más cerca del nuevo objeto.

El módulo *Control* genera señales que le permiten seguir los diferentes caminos locales y corregir las desviaciones respecto a los mismos de la trayectoria estimada (que se presenta en la gráfica (a) de la figura C5-32).

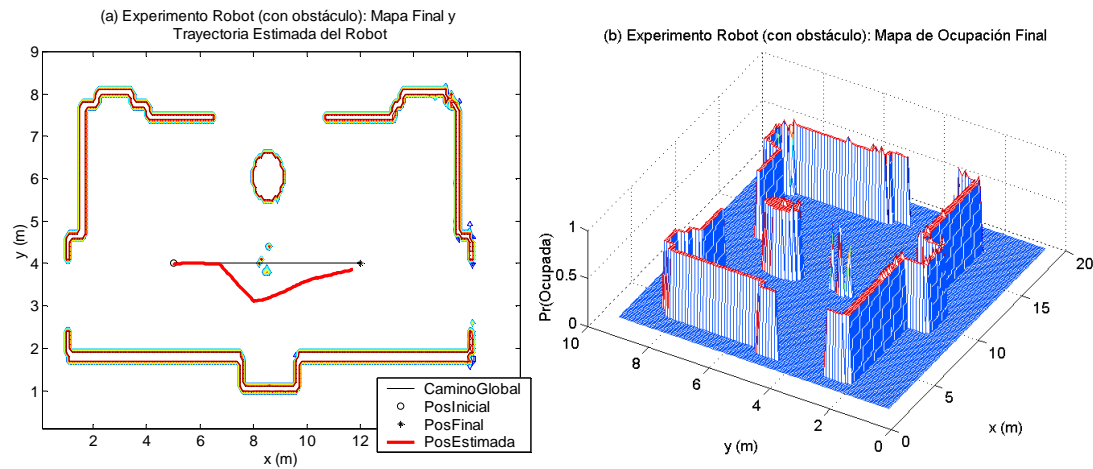


Figura C5-32. Mapa probabilístico de ocupación final y trayectoria estimada del robot para el experimento con todos los sensores funcionando y con un obstáculo en el mapa de objetos usado en la simulación. En la gráfica (a) se representa el mapa probabilístico final de celdas como un plano, el punto de partida del robot (PosInicial), el solicitado como posición objetivo final (PosFinal), el camino global propuesto inicialmente por el módulo *Planificador*, y la trayectoria estimada por el algoritmo IFAsynMixto. En la gráfica (b) se presenta el mapa probabilístico final en tres dimensiones: los valores del eje z representan la probabilidad de ocupación de cada celda (x,y) del mapa.

En la figura C5-33 se representa, para cada instante de aplicación de la señal de control, que sucede con la brújula y los encoders. Obviaremos el análisis del comportamiento de estos dos sensores, ya que como se puede observar en la gráfica, es básicamente similar al de los ejemplos anteriores.

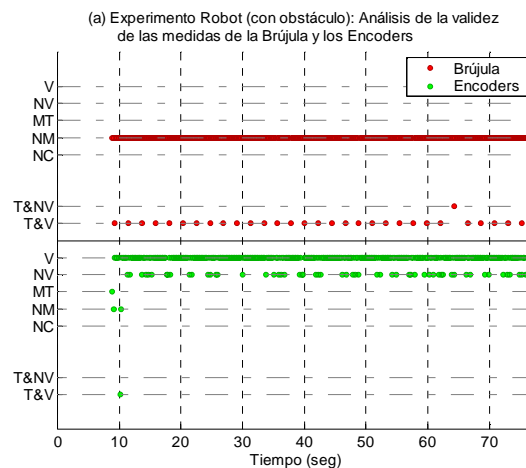


Figura C5-33. Análisis de la información sensorial proporcionada por los encoders y la brújula en el experimento con todos los sensores funcionando y con un obstáculo en el mapa de objetos de simulación. En cada gráfica se presenta se representa para cada periodo de muestreo de la señal de control la suerte sufrida por las medidas (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, T&NV tarde y no válida, T&V tarde y válida).

En las gráficas de la figura C5-34 se representa lo acontecido con los sensores de ultrasonidos que han sido capaces de detectar algún objeto (todos ellos salvo el sensor de ultrasonidos 7).

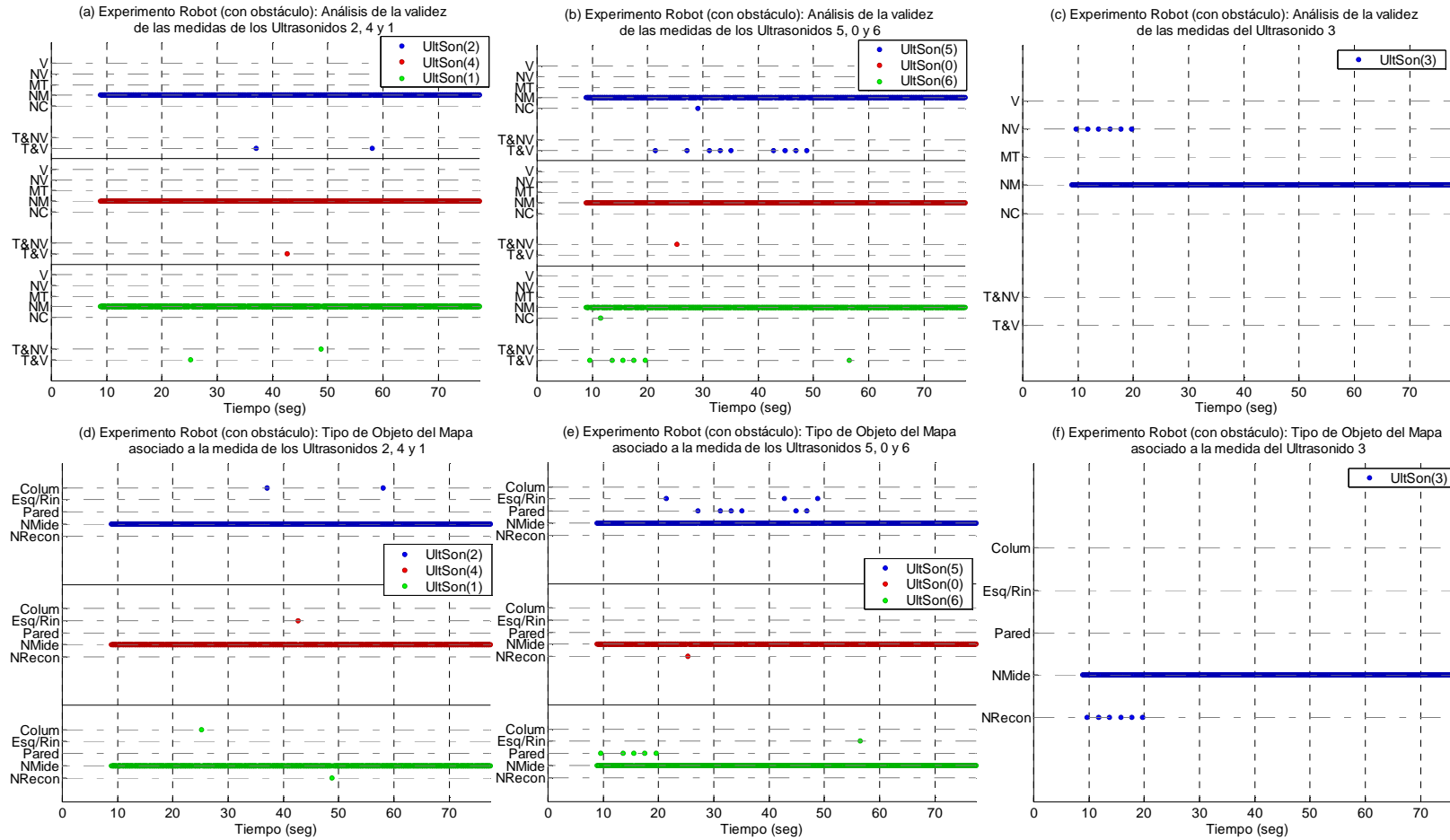


Figura C5-34. Análisis, para el experimento de mapa de objetos de simulación con obstáculo, de la validez de las medidas de los ultrasonidos y del tipo de objeto asociado por el sistema ultrasonidos + mapa. En las gráficas (a), (b) y (c), para cada periodo de muestreo de la señal de control, se representa la suerte sufrida por las medidas (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, T&NV tarde y no válida, T&V tarde y válida). En las gráficas (d), (e) y (f) se representa si no hay una medida procesable (NMide), y si la hay pero el sistema ultrasonidos + mapa no es capaz de asociarla a ningún objeto (NRecon), o la asocia a una Pared (Pared), Columna (Colum), o esquina o rincón (Esq/Rin).

Los resultado extraíbles del análisis de las gráficas de figura C5-34, en el que se tiene en cuenta la posición (x, y, θ) en la que se encuentra robot (representada en la gráfica (a) de la de la figura C5-35) cuando se realiza la asociación de medidas de distancia de ultrasonidos al mapa de objetos inicial utilizado por el algoritmo IFAsynMixto (sin obstáculo, representado en la gráfica (a) de la figura C5-24), la posición de los ultrasonidos sobre el robot (gráfica (e) de la figura C5-30) y la posición de los objetos del mapa utilizado en la simulación de las medidas de los sensores (gráfica (b) de la figura C5-24), son los siguientes:

- Comparando las parejas de gráficas de la figura C5-34 correspondientes al mismo conjunto de sensores se comprueba que en este ejemplo todas las medidas que son asociadas a algún tipo de objeto del mapa (*Colum*, *Esq/Rin*, *Pared*) son a continuación validadas (*T&V*) por el test chi-cuadrado. Las medidas no válidas (*T&NV* y *NV*) coinciden con aquellas que no han sido asociadas a ningún objeto del mapa (*NRecon*).
- Las medidas no reconocibles (y por lo tanto no válidas) corresponden al obstáculo no recogido en el mapa inicial del algoritmo IFAsynMixto. Mientras el robot sigue el plan inicial del robot (paralelo al eje x), el sensor de ultrasonidos 3, situado en el frente del robot, detecta el obstáculo. Tan pronto como el robot gira hacia la pared para evitar el obstáculo y empieza a trasladarse, el sensor de ultrasonidos 0 detecta el un nuevo pedazo del obstáculo, relativamente próximo al robot, lo que hace que el *Planificador* detenga al robot durante unos segundos, como medida de seguridad ante un posible objeto en movimiento. Después de reanudar su marcha, y una vez que el robot ha superado el obstáculo y cambia su orientación para acercarse al objetivo final, el sensor de ultrasonidos 1 vuelve a detectar el obstáculo.
- Las medidas válidas que se corresponden a una pared son aportadas por el ultrasonido 6, al inicio del recorrido, cuando el robot se mueve según el plan inicial, y por el sensor 5 cuando el robot gira para evitar el obstáculo y permanece parado durante unos segundos, y cuando vuelve a girar una vez superado el obstáculo. Las medidas corresponden a paredes son paralelas al eje x , por lo que aportarán más información sobre el estado y que sobre el x , reduciendo la varianza del estado y y no la del x .
- El resto de las medidas válidas son asociadas a la columna y a las esquinas o rincones existentes en el mapa de objetos inicial. Como este tipo de medida aporta información tanto sobre el estado x como sobre el estado y , reducirán la varianza de ambos estados.

En las gráficas de la siguiente figura se presenta la evolución del valor de los estados (x, y, θ) y la de su varianza. Para cada estado/varianza se representan dos curvas:

- *Inicio* presenta el valor del estado (varianza) calculado antes de realizar la predicción que le llevará al próximo instante de muestreo y de recibir la información asociada a instantes de muestreo anteriores que llega después de que se realice la predicción.
- *Final* corresponde al valor del estado (varianza) una vez que se ha asimilado toda la información del instante temporalmente retardada.

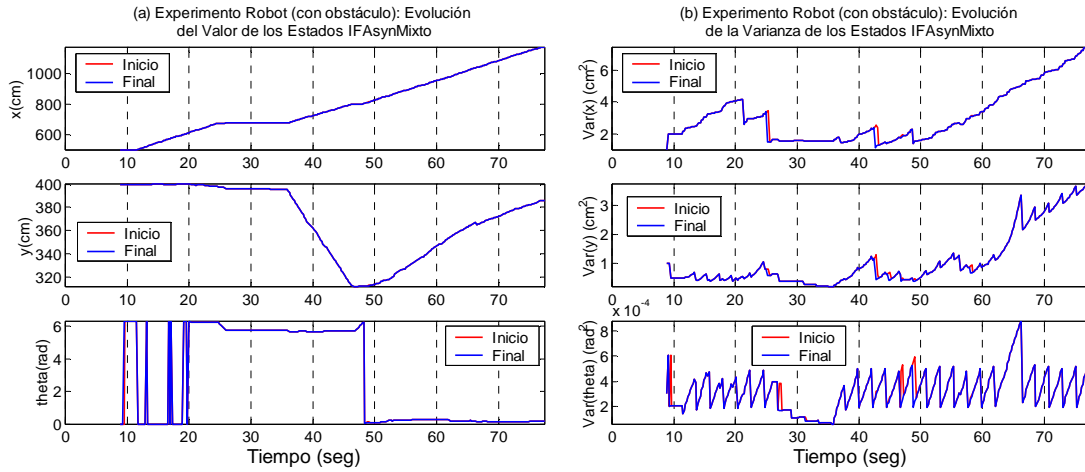


Figura C5-35. Evolución del valor estimado de los estados (x , y , θ) y su varianza para el experimento con todos los sensores funcionando y con un mapa de objetos con obstáculo para la simulación.

La diferencia de las dos curvas para los estados x e y no resulta observable en las gráficas, ya que las correcciones introducidas por la asimilación de la información sensorial retardada no resultan significativas en este caso. Sin embargo, para el estado θ se puede apreciar un cambio significativo al inicio, debido a la discontinuidad $0-2\pi$. La existencia de ese cambio pone de manifiesto la importancia del recálculo de la información sensorial asociada a la brújula en el caso de la asimilación retardada de las medidas de otros sensores. Si en la asimilación se pasa al lado opuesto de la discontinuidad de aquel en el que se encuentra la información asociada a la brújula, el resultado de la asimilación de toda la información resultará en un valor erróneo, tal y como se justifica detalladamente en el apéndice V.

En las curvas de las varianzas, se observa siempre una reducción de la segunda (Final), ya que la asimilación de información sensorial (aunque llegue retardada) reduce el desconocimiento que se tiene sobre el estado de sistema. Además,

- El valor constante en la varianza de los estados x e y durante unos segundos (de los 25 a los 35) se debe a que el robot permanece parado y el aumento debido a la predicción es por lo tanto menor. Durante el mismo tiempo, el error de orientación disminuye, ya que la brújula proporciona medidas.

- Las reducciones de varianza de θ se encuentran asociados a los instantes de asimilación de las medidas de la brújula (el pico más significativo en la varianza de θ es debido al rechazo de una de sus medidas) y de los ultrasonidos.
- Las reducciones de varianza de x e y se deben a la asimilación de las medidas de ultrasonidos (en el estado x causadas por la detección de columnas y esquinas, mientras que en el estado y originadas por la detección de cualquier tipo de objetos).
- El crecimiento de la varianza de los estados x e y después de la última medida válida de ultrasonidos (próxima a los 60 segundos) se debe a la falta de medidas de este tipo de sensor que permitan posicionar el robot de forma absoluta.

Con el experimento se ha comprobado como las medidas de los ultrasonidos no asociables a ningún objeto del mapa inicial utilizado por los modelos de ultrasonidos + mapa del algoritmo IFAsynMixto son rechazadas por el algoritmo de asociación, y por lo tanto, no utilizadas por el sistema de fusión para estimar la posición del robot. Sin embargo, estas medidas si son utilizadas por el algoritmo encargado de estimar el estado del entorno, para permitir que el robot pueda reaccionar ante los cambios observados en el entorno. La reactividad del sistema, que le permite esquivar los objetos no recogidos en el mapa inicial, provoca el cálculo de nuevos caminos que apartan al robot de los objetos del mapa detectable por los sensores de ultrasonidos, hecho que incrementa el desconocimiento de la posición real del robot. Este problema puede ser solucionado incrementando el número de objetos que le permitan posicionarse al robot de forma absoluta o incorporando nuevos sensores al robot cuya información pueda ser utilizada por el algoritmo IFAsynMixto para estimar mejor la posición del robot.

C5.2.2. Resultados experimentales sobre el robot real

Los experimentos realizados sobre el robot descrito en el apéndice III permiten analizar el comportamiento del algoritmo IFAsynMixto cuando es utilizado como parte de lazo de control para estimar la posición de un móvil, y estudiar la completitud y exactitud de los modelos de movimiento y sensoriales utilizados durante la etapa de simulación. La arquitectura de control utilizada para estos experimentos es la misma que para los experimentos simulados, salvo por la utilización en el módulo *ElRobot* de las funciones necesarias para interactuar con el robot (en vez de simular su comportamiento) y la modificación de un conjunto de parámetros que nos permiten sintonizar el comportamiento del sistema sensorial y el algoritmo IFAsynMixto: frecuencia de muestreo de los sensores, los retardos mínimos introducidos en la comunicación por el módulo *ElRobot*, covarianzas de los modelos de movimiento y transición, etc. Los principales cambios, que se recogen en las tablas C5-3 y C5-4, son debidos a que:

- Los sensores de infrarrojos y el sistema de visión se encuentran desactivados, ya que su funcionamiento aumenta la carga computacional de la arquitectura de control, sin proporcionar información al algoritmo IFAsynMixto, centro de los análisis de los resultados de este capítulo. La desactivación de los infrarrojos y el sistema de visión, hacen que el módulo *Entorno* actualice el estado del entorno, almacenado como un mapa probabilístico de ocupación, únicamente con la información procedente de los sensores de ultrasonidos, por lo que es conveniente utilizar una frecuencia de muestreo superior que la utilizada en los experimentos simulados.
- El error de las medidas de los ultrasonidos no es constante, sino que depende de la distancia medida z debido a la directividad asociada a los sensores. Por lo tanto, se incluye un término dependiente de z y de la directividad en la varianza de estos sensores.
- A la covarianza de la brújula se le había asociado durante los experimentos simulados la precisión proporcionada por el fabricante. Ésta no tiene en cuenta las fluctuaciones del campo magnético existentes en un edificio, que hacen conveniente reducir el peso que las medidas de este sensor tienen en los resultados del algoritmo IFAsynMixto. Esto se consigue aumentando la varianza asociada al sensor. De esta forma la información de las medidas de la brújula será considerada menos fiable, aunque el algoritmo podrá utilizar un mayor número de medidas, al ser más fácil que superen el test de validación chi-cuadrado.
- La varianza asociada al modelo de movimiento en los experimentos simulados no tenía en cuenta que cuando el robot gira debido a la aplicación de señales de control diferentes sobre cada rueda, el error en la orientación es mayor que en los casos en los que se le aplica la misma velocidad a ambas ruedas.
- La posición inicial utilizada durante los experimentos simulados en el algoritmo IFAsynMixto y en el módulo *ElRobot* es la misma, por lo que se le puede asignar valores pequeños a la matriz inicial de covarianza del estado. En los experimentos reales, el valor con el que se inicializa el algoritmo IFAsynMixto tiene que ser lo más cercano posible a la posición real sobre la que se sitúa el robot. El desconocimiento de la diferencia existente entre la posición real y la de inicialización del algoritmo IFAsynMixto se indica con una matriz inicial de covarianza del estado significativamente mayor, apreciable en las gráficas de varianzas recogidas durante los resultados experimentales presentados en las siguientes secciones.

Los valores y expresiones de las covarianzas variables de las matrices \mathbf{Q}_t y $\mathbf{R}_{s,t}$ del apéndice V para los experimentos reales, así como los nuevos periodos de muestreo y retardos mínimos utilizados, también se recogen en las dos tablas siguientes. A la hora de analizar los resultados de los experimentos es importante tener siempre presente que las medidas de la brújula y de buena parte de los ultrasonidos llega al algoritmo IFAsynMixto de forma retardada (y desordenadas entre

si), y por lo tanto, se está explotando la capacidad de este algoritmo para trabajar con información desordenada.

Covarianzas de los Modelos de			
Movimiento	Brújula	Encoders	Ultrasonidos
$\text{var}_{\Delta l}(v_L, v_R) = (0.02 \cdot a)^2$ $\text{var}_{\Delta \theta}(v_L, v_R) = (2\pi \cdot a \cdot b / 180)^2$ $\text{cov}_{\Delta l \Delta \theta}(v_L, v_R) = 0$ $a = \begin{cases} 0.1 & \text{si } (v_L = 0) \wedge (v_R = 0) \\ 1 & \text{c.c.} \end{cases}$ $b = \begin{cases} 4 & \text{si } (v_L \neq v_R) \\ 1 & \text{c.c.} \end{cases}$	$\text{var}_{\text{brújula}} = (3\pi / 180)$	$\text{var}_{N_L} = (0.01)^2$ $\text{var}_{N_R} = (0.01)^2$	$\text{var}_{\text{sonar}} = (0.02 + a)^2$ $a = z \cdot \cos(10\pi / 180)$

Tabla C5-3. Valores de las matrices de covarianza de los modelos de movimiento y de medida utilizados por los sensores activados durante los experimentos reales. Los valores de $\text{var}_{\Delta \theta}$ y $\text{var}_{\text{brújula}}$ se recogen en rad^2 , mientras que los de $\text{var}_{\Delta l}$, var_{N_L} , var_{N_R} y $\text{var}_{\text{sonar}}$ se presentan en m^2 .

Encoders		Brújula		Ultrasonidos	
Periodo (seg)	Retardo (seg)	Periodo (seg)	Retardo (seg)	Periodo (seg)	Retardo (seg)
0.2	--	2.0	1.0	1.0	0.0 a 0.6

Tabla C5-4. Periodos de muestreo y retardos de comunicación mínimo asociado de los sensores activados en los experimentos reales.

Por último, destacar que los experimentos reales han sido realizados sobre dos espacios diferentes, un pasillo y un hall (similar al de los experimentos simulados pero con distintos obstáculos), para los que ha sido necesario generar el mapa de objetos utilizado por el algoritmo IFAsynMixto y los mapas probabilísticos de ocupación inicial utilizados por los módulos *Entorno* y *Planificador*. Al ser experimentos reales, el módulo *ElRobot* no necesita disponer de ningún mapa de objetos, ya que éste solamente era utilizado para simular las medidas de los sensores de ultrasonidos, infrarrojos, y el sistema de visión. Estos mapas se presentan en las siguientes secciones, junto con los resultados obtenidos para cada uno de los experimentos reales recogidos en esta tesis.

Experimento real en un pasillo: desplazamiento del robot a lo largo del mismo

El primero de los experimentos reales consiste en colocar el robot en un pasillo, y solicitar a la arquitectura de control que éste se mueva en línea recta a lo largo del mismo. El mapa inicial de objetos del pasillo, representado en la siguiente figura, se ha orientado con las paredes paralelas al eje x. Los sensores de ultrasonidos situados en los dos laterales del robot pueden determinar la distancia del robot a las paredes, y por lo tanto, disminuir la covarianza asociada al estado y cada vez que se produce una medida de cualquiera de los dos sensores de ultrasonidos. Una de las paredes del pasillo está franqueada por varias puertas, que permanecerán cerradas durante el

experimento, cuyos marcos serán modelados como esquinas/rincones en el mapa inicial de objetos utilizados en el sistema ultrasonidos + mapa del algoritmo IFAsynMixto.

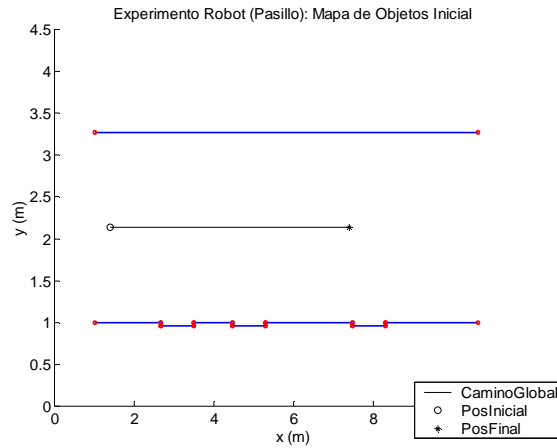


Figura C5-36. Mapa de Objetos utilizado por el sistema ultrasonidos + mapa del algoritmo IFAsynMixto durante el experimento real en el pasillo, junto con las posiciones inicial y final del robot, y el camino global planificado inicialmente. En azul se representan las paredes, y en rojo las esquinas o rincones.

En este experimento no se coloca ningún objeto en el pasillo, por lo que el módulo *Entorno* no modificará de forma significativa el mapa probabilístico de ocupación inicial, cuyos valores iniciales se representan en la gráfica (a)- bajo la forma de plano- y en (c)- en tres dimensiones- de la figura C5-37. Los cambios producidos a lo largo del experimento, como consecuencia de la utilización de la información proporcionada por los sensores de ultrasonidos, se producen en torno a las paredes (y puertas) del pasillo y se observan en las gráficas (b) y (d) como incrementos de la probabilidad de ocupación.

La falta de cambios significativos en el mapa de ocupación en las zonas que inicialmente tenían una probabilidad baja, hacen que el *Planificador*, en sus labores de vigía, permita al robot seguir con el camino global calculado al inicio de la ejecución del programa (y que se encuentra representado en la figura C5-36 y en las gráficas (a) y (b) de la figura C5-37). Mientras tanto el módulo *Control* genera las órdenes para seguir el camino y corregir las desviaciones respecto al mismo de la trayectoria estimada (representada en la gráfica (b) de la figura C5-37).

En las gráficas de la figura C5-38 se presentan los resultados del algoritmo IFAsynMixto, junto al funcionamiento de los sensores, la suerte sufrida por las medidas durante el proceso de validación, y para las medidas del sistema ultrasonidos + mapa el tipo de objeto asociado a cada una de las medidas en la que se ha detectado un objeto.

La suerte sufrida por las medidas de la brújula (que llegan al algoritmo IFAsynMixto temporalmente retardadas) y de los encoders, representada en la gráfica (a) de la figura C5-38, es similar a la observada durante los experimentos simulados. En este caso, todas las medidas de la brújula son válidas (*T&V*), así como la mayor parte de las medidas de los encoders (*V*).

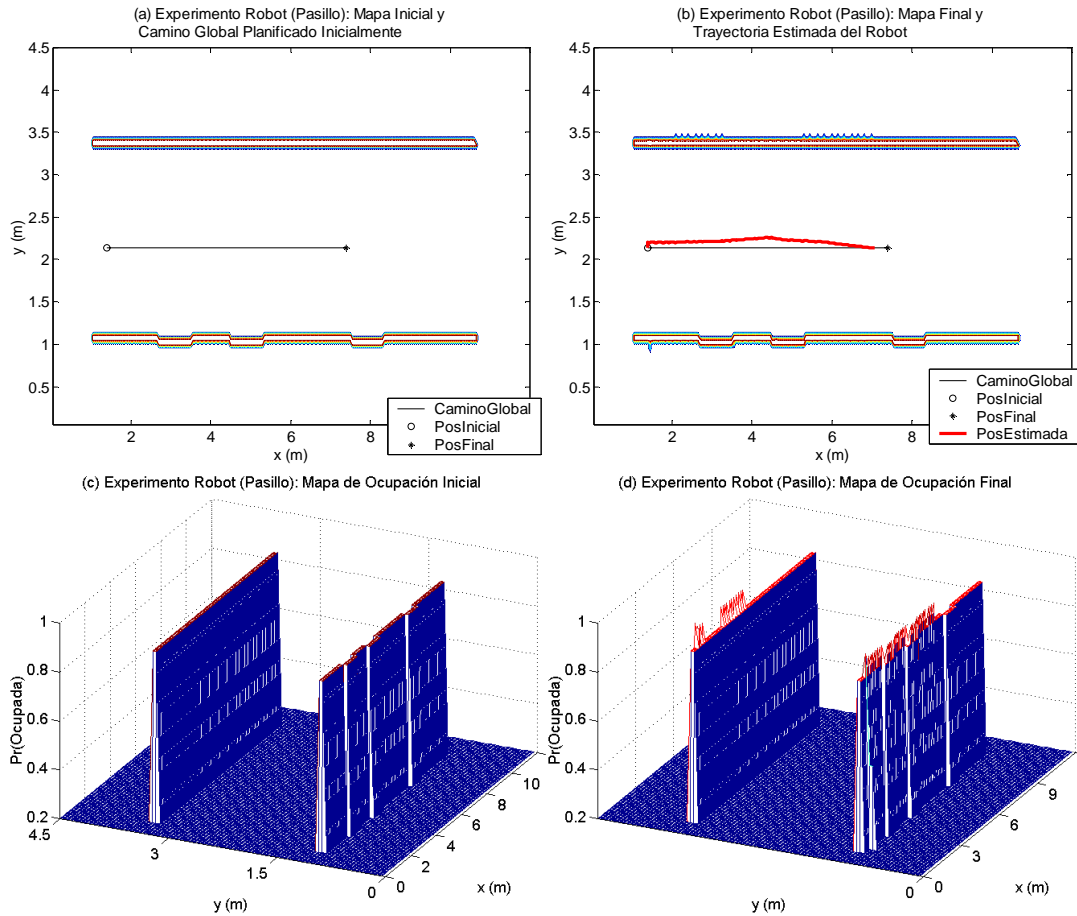


Figura C5-37. Mapas probabilísticos de ocupación inicial y final, junto con el punto de partida y objetivo final del robot, y la trayectoria estimada, para el experimento real en el pasillo. En (a) se representa el mapa probabilístico de celdas como un plano, el punto de partida del robot (PosInicial), el solicitado como posición objetivo final (PosFinal), y el camino global propuesto inicialmente por el módulo *Planificador*. En (b) se muestra el mismo mapa al final de la trayectoria, junto con la trayectoria estimada. En (c) y (d) se recogen los mapa probabilísticos de celdas inicial y final en tres dimensiones: los valores del eje z corresponden a la probabilidad de ocupación de cada celda (x,y).

En la gráfica (b) de la figura C5-38 se observa que los dos ultrasonidos situados en los laterales del robot, que se encuentran orientados ortogonales a las dos paredes del pasillo, detectan objetos en todos los instantes en los que muestrean, aunque dos de las medidas proporcionadas por el sensor 6 del cinturón de ultrasonidos, no son asignables a ningún instante de aplicación de la señal de control (NC) por lo que no son procesables por el algoritmo IFAsynMixto. Las medidas restantes llegan al algoritmo IFAsynMixto retardadas, y son asimiladas ya que las dos etapas del proceso de validación de las medidas de ultrasonidos consideran que son válidas. En la gráfica (c) se observa que durante la primera etapa del proceso de validación, realizada en el algoritmo de asociación de las medidas a objetos del mapa, se le asignan a las medidas paredes (*Pared*), y rincones o esquinas (*Esq/Rin*), correspondientes respectivamente a las paredes o puertas, y a los marcos de las puertas.

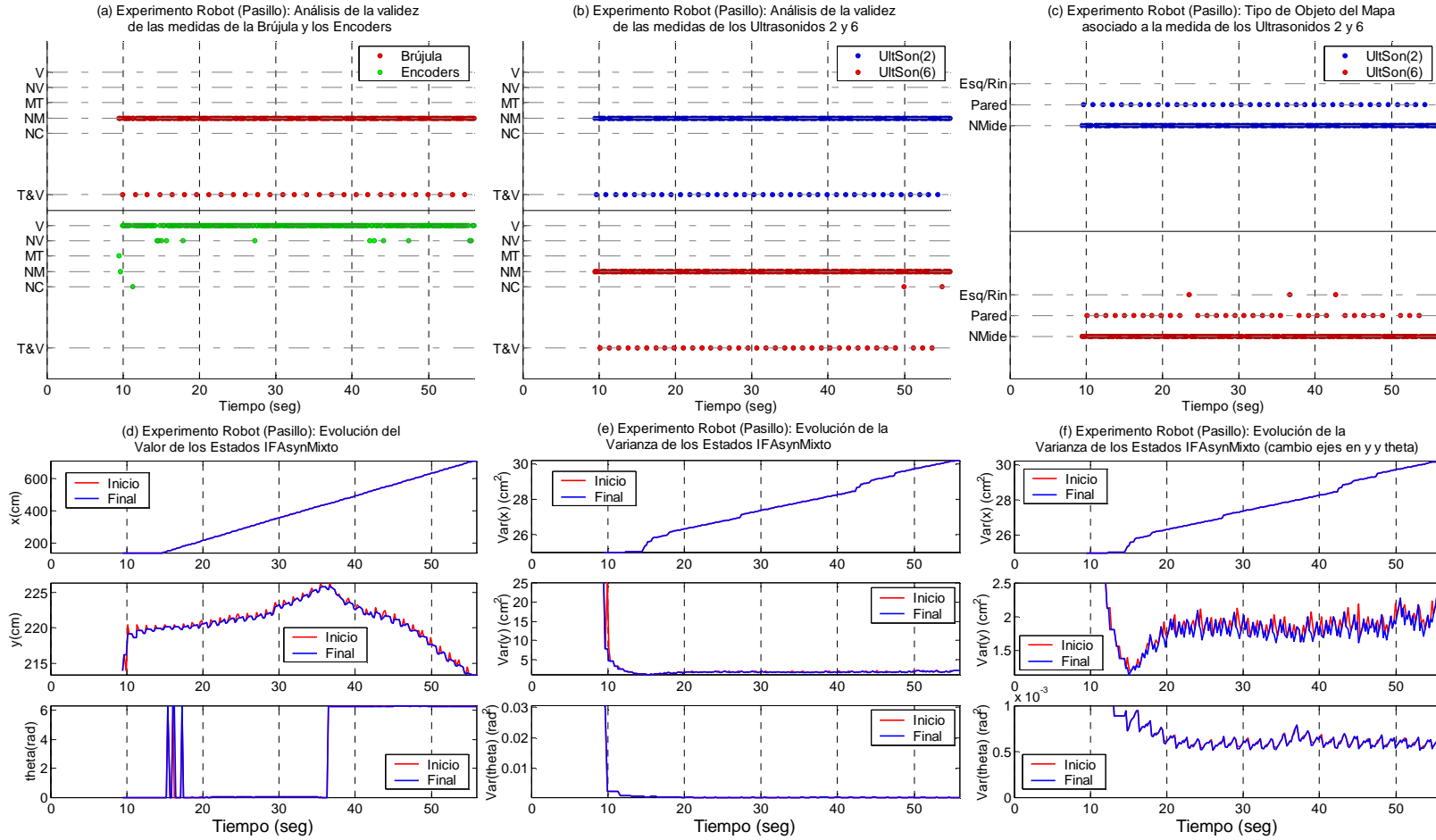


Figura C5-38. Resultados del experimento real en el pasillo. En las gráficas (a) y (b) se muestra que ocurre con las medidas (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, T&V tarde y válida). En la gráfica (c), para los ultrasonidos, se representa si no hay una medida procesable (NMide), y si la hay pero el sistema ultrasonidos + mapa no es capaz de asociarla a ningún objeto (NRecon), o la asocia a una Pared (Pared), o esquina o rincón (Esq/Rin). En las gráficas (d), (e) y (f) se muestra la evolución del valor estimado de los estados (x , y , θ) y su varianza.

Como el robot se mueve según la orientación del eje x la información de las medidas de ultrasonidos permite reducir el valor de la varianza del estado y , y no del x (representadas ambas en la gráfica (e) de la figura). Esto ocurre a pesar de que dos de las medidas corresponden a esquinas o rincones, ya que la profundidad de los marcos es muy pequeña y estos objetos son detectados cuando el robot se mueve paralelo a las paredes.

Para los valores de cada estado y de sus varianzas se representan dos curvas, una asociada al inicio del proceso de asimilación de la medida para cada instante de muestreo y otra correspondiente al final de la asimilación de todas las medidas asociadas a dicho instante y a instantes anteriores que llegan al algoritmo IFAsynMixto de forma temporalmente retardada.

En este caso, a simple vista se observan las correcciones que se producen en el valor y varianza del estado y . Éstas son debidas a que las medidas de los ultrasonidos que llegan al algoritmo de forma retardada son válidas. Las asociadas a cada uno de los laterales del robot compiten por determinar la posición del mismo, y corrigen la posición y del robot, y disminuyen la varianza de dicho estado. Además, las medidas de la brújula (retardadas y válidas) también mejoran la varianza de θ .

Por último, es importante aclarar que la desviación que se observa en la trayectoria estimada del robot ha sido observada durante su desplazamiento real. Las medidas proporcionadas por los sensores permiten que el algoritmo IFAsynMixto detecte la desviación inicial, y que el algoritmo de control intente corregirla.

Experimento real en el hall: desplazamiento del robot y esquiva de obstáculos

El segundo de los experimentos reales consiste en colocar el robot en un hall y solicitar a la arquitectura de control que el robot se mueva en línea recta a lo largo del mismo, de forma que se encuentre con dos obstáculos no recogidos en el mapa de entorno inicial ni en el mapa utilizado en el sistema ultrasonidos + mapa del algoritmo IFAsynMixto. En la gráfica (a) de la figura C5-39 se representa el mapa de objetos utilizados en el algoritmo IFAsynMixto, mientras que en la gráfica (b) se representa el mismo mapa de objetos junto con la disposición de los objetos utilizados como obstáculo: una caja rectangular y un cenicero cilíndrico. El primero de los obstáculos se sitúa sobre el camino global que el *Planificador* calculará inicialmente, mientras que el segundo se coloca en las proximidades del entorno por el que se prevé que la arquitectura de control dirigirá al robot para esquivar el primero de los objetos. Este experimento es similar al realizado con obstáculos durante las simulaciones, diferenciándose principalmente en la no existencia de una de las columnas en el mapa inicial, y la utilización de dos objetos imprevistos en vez de uno.

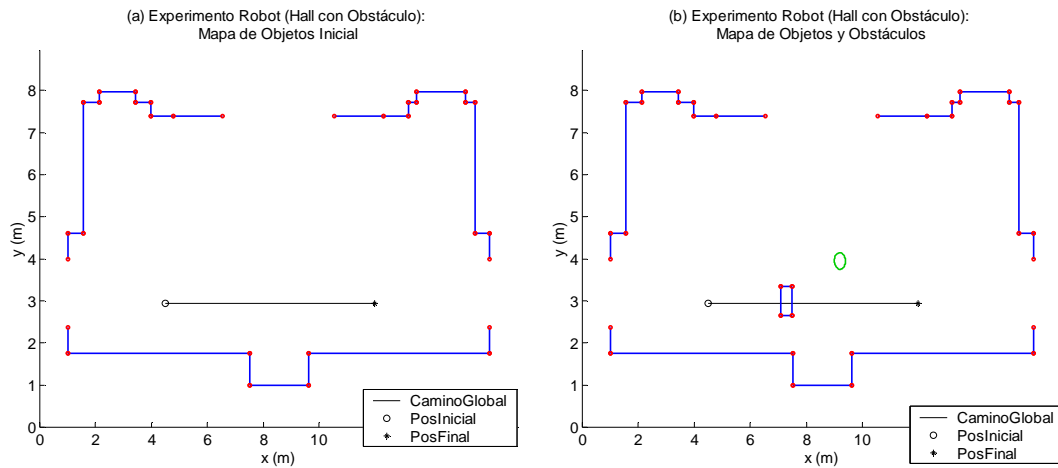


Figura C5-39. Mapa de los objetos utilizados durante los experimentos reales en el hall, junto con las posiciones inicial y final del robot, y el camino global planificado inicialmente. En azul se representan las paredes, en rojo las esquinas y los rincones, y en verde los objetos cilíndricos. El sistema ultrasonidos + mapa de objetos incorporado en el algoritmo IFAsynMixto utiliza el mapa de la gráfica (a). En la gráfica (b) se muestran dos objetos adicionales, que representan la posición de los dos obstáculos reales situados en el hall.

La existencia de dos obstáculos hace que el módulo *Entorno* modifique el mapa probabilístico de ocupación inicial (cuyos valores iniciales se representan en la gráfica (a)- bajo la forma de plano- y en (c)- en tres dimensiones- de la figura C5-40) según los ultrasonidos van detectando la presencia de nuevos objetos o confirmando la existencia de los conocidos inicialmente. El mapa probabilístico final, en el que se recogen las probabilidades de ocupación del entorno una vez finalizado el experimento, se muestra en las gráficas (b) y (d) de la misma figura. Comparando los mapas finales e iniciales se observa un crecimiento de la probabilidad de ocupación en las zonas en las que se han colocado los obstáculos (la caja rectangular es observada por tres de sus laterales) y en las paredes próximas a la trayectoria del robot. La orientación de los mapas de ocupación en tres dimensiones de este ejemplo es diferente a las de los de la sección de simulación.

Los cambios significativos en el mapa de ocupación relativos a zonas con una baja probabilidad inicial hacen que el módulo *Planificador* calcule un conjunto de caminos locales alternativos, para esquivar la presencia de los nuevos objetos, según éstos son detectados y el robot se está desplazando. La parte posterior de la caja rectangular es observada cuando el robot se encuentra muy cerca de la misma, y por lo tanto, el *Planificador* y el *Navegante* solicitarán al robot que permanezca parado durante un periodo de tiempo, como medida de seguridad para poder observar durante más tiempo el objeto y comprobar si éste se mueve o no. Mientras tanto, el módulo *Control* generará las órdenes para permanecer parado el tiempo necesario, seguir los nuevos caminos locales y corregir las desviaciones respecto a los mismos de la trayectoria estimada (representada en la gráfica (c) de la figura C5-40).

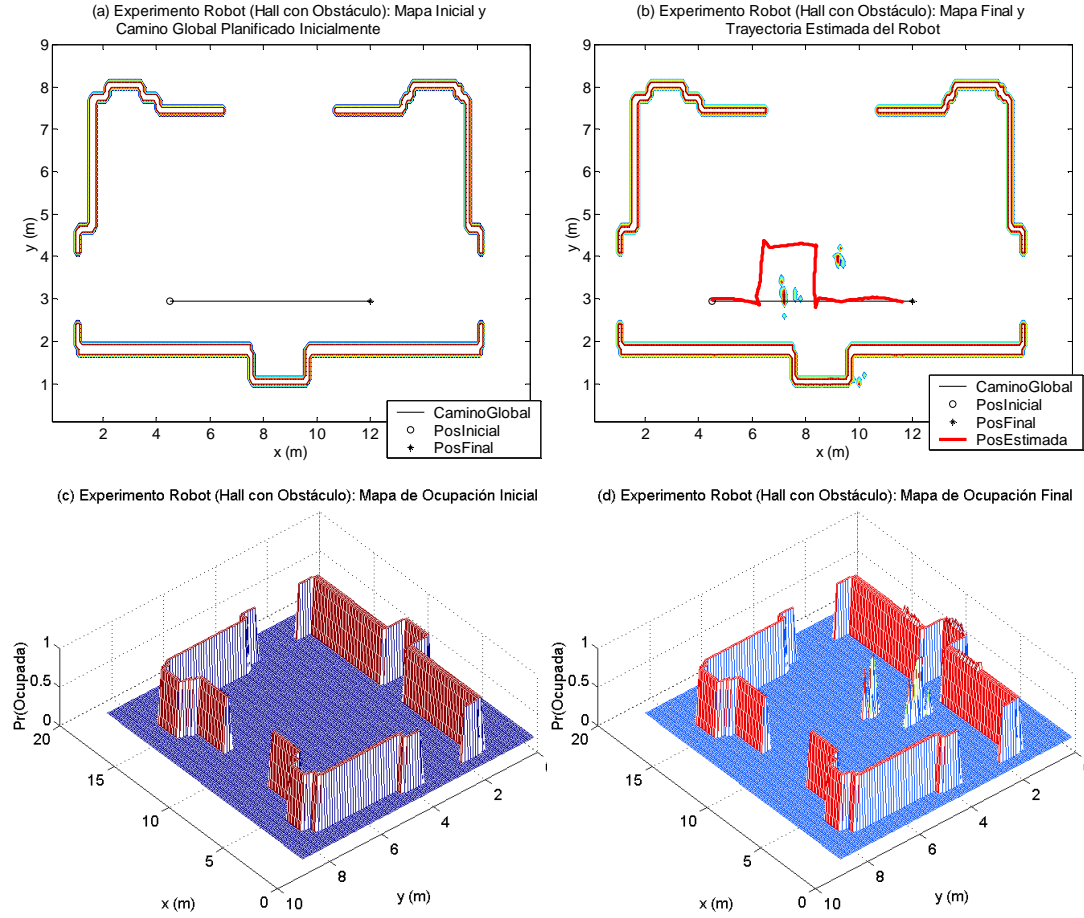


Figura C5-40. Mapas probabilísticos de ocupación inicial y final, junto con el punto de partida y objetivo final del robot, y la trayectoria estimada, para el experimento real en hall con obstáculos. En (a) se representa el mapa probabilístico de celdas como un plano, el punto de partida del robot (PosInicial), el solicitado como posición objetivo final (PosFinal), y el camino global propuesto inicialmente por el *Planificador*. En (b) se muestra el mismo mapa al final de la trayectoria, junto con la trayectoria estimada. En (c) y (d) se recogen los mapa probabilísticos de celdas inicial y final en tres dimensiones: los valores del eje z corresponden a la probabilidad de ocupación de cada celda (x,y).

El análisis de lo sucedido en el algoritmo IFAsynMixto se representa en las gráficas de las siguientes figuras: el funcionamiento de los sensores junto con la suerte sufrida por sus medidas durante el proceso de validación, que consta de dos etapas en el caso del sistema de ultrasonidos + mapa, se representa las figuras C5-41 y C5-42, mientras que los valores estimados para los estados (x, y, θ) y sus varianzas se muestran en la figura C5-43.

Casi todas las medidas de la brújula y de los encoders son válidas, en el primer caso llegan al algoritmo retardadas ($T \& V$) y en el segundo, casi todas, son recibidas por el algoritmo IFAsynMixto antes de que éste realice una nueva etapa de predicción. Esta información se representa en la gráfica de la figura C5-41.

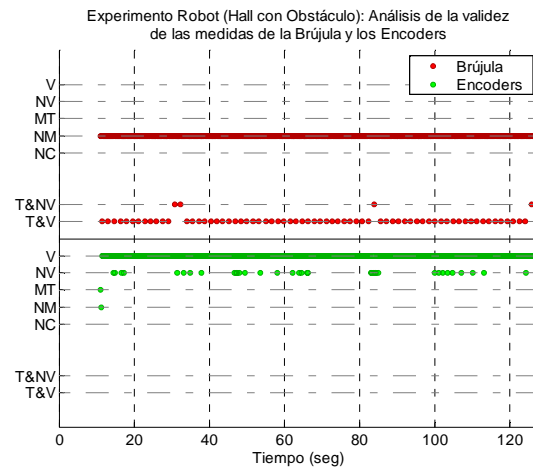


Figura C5-41. Análisis de la información sensorial proporcionada por los encoders y la brújula en el experimento real en el hall con obstáculos. En cada gráfica se presenta para cada periodo de muestreo de la señal de control la suerte sufrida por las medidas (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, T&NV tarde y no válida, T&V tarde y válida).

Para realizar el análisis de lo que sucede con las medidas de ultrasonidos hay que tener en cuenta que aunque los sensores miden cada segundo sólo proporcionan una medida cuando se detecta un objeto. Las medidas son originadas por todos los objetos colocados en el mapa próximos al robot, tanto por aquellos conocidos a priori por los algoritmos del módulo *Entorno y Posicion* (IFAsynMixto) como por los obstáculos desconocidos. El algoritmo de asociación del sistema ultrasonidos + mapa se encarga de discernir entre el tipo de objetos que se asigna a cada medida, y en caso de no corresponder a ningún objeto del mapa, de rechazarla como información sensorial no válida para el algoritmo IFAsynMixto. Esta información se recoge, para los diferentes sensores, en las gráficas (d), (e) y (f) de la figura C5-42. Las medidas asociadas a objetos son sometidas a un test de validación chi-cuadrado, cuyos resultados se recogen en las gráficas (a), (b) y (c) de la misma figura, considerándose en las mismas medidas no válidas (NV o T&NV) aquellas que no han superado ninguna de las dos etapas del proceso de validación. La interpretación de los resultados se realiza teniendo en cuenta la disposición de los sensores en el robot (representada en la gráfica (e) de la figura C5-30) y la posición y orientación del robot en cada instante (cuyos valores estimados se representan respectivamente en las gráficas (b) y (a) de las figuras C5-40 y C5-43). De toda esta información se concluye que:

- Las medidas del sensor de ultrasonidos 2 corresponden al obstáculo cilíndrico por lo que no son reconocibles por el algoritmo de asociación, y por lo tanto, no son asimiladas por el algoritmo IFAsynMixto. Lo mismo sucede con las medidas del sensor de ultrasonidos 1.
- Las dos primeras medidas del ultrasonidos 4 se corresponden a cada uno de los obstáculos (la primera a la caja y la segunda al obstáculo cilíndrico) por lo que no son asociables a ningún objeto. Las cuatro restantes se corresponden al entrante inferior del hall. Dos de ellas son rechazadas por el algoritmo de asociación, y las otras dos asignadas a objetos del mapa. Estas últimas son validadas por el test chi-cuadrado y asimiladas por el algoritmo.

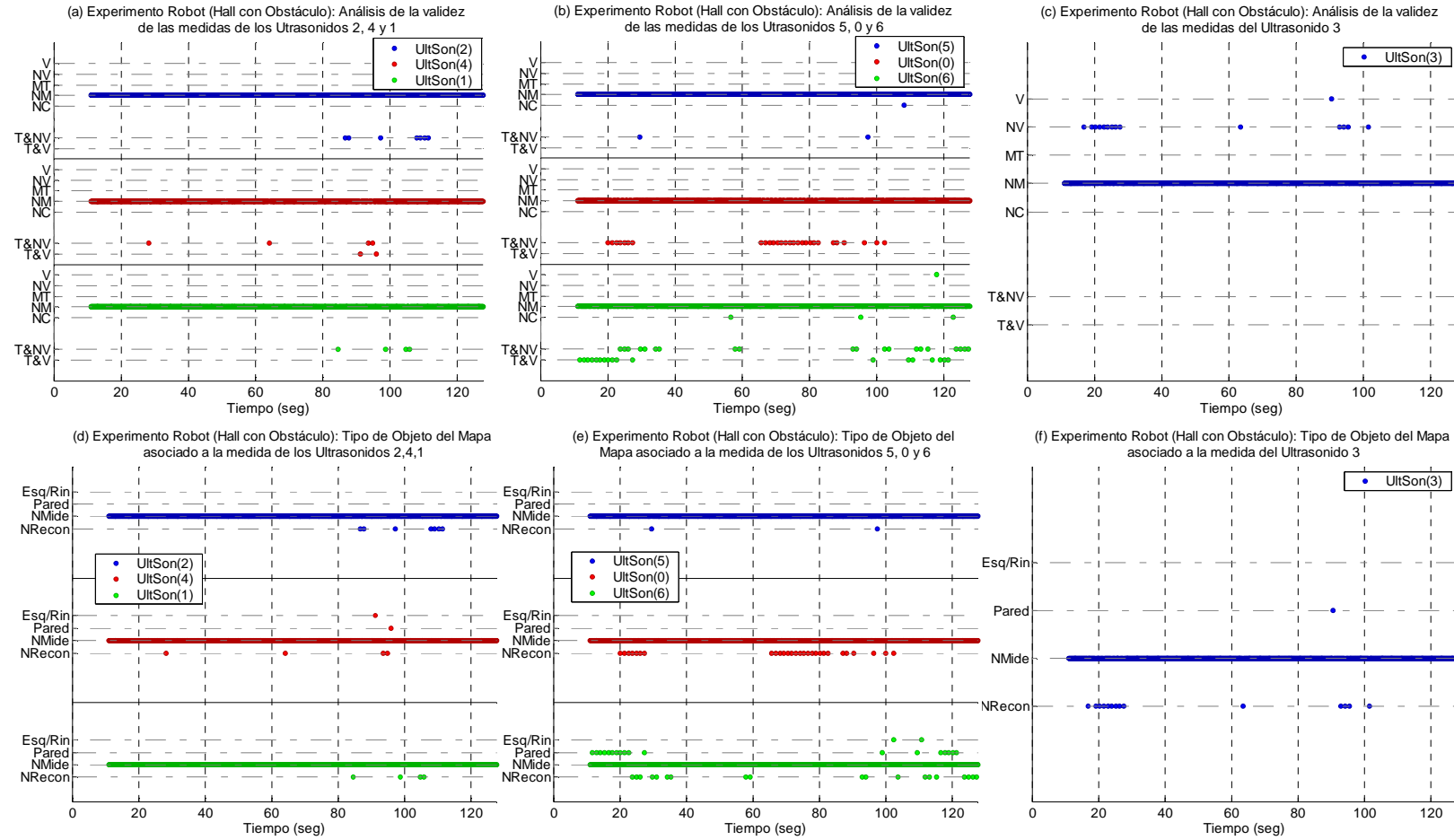


Figura C5-42. Análisis, para el experimento real del hall con obstáculos, de la validez de las medidas de los ultrasonidos y del tipo de objeto asociado por el sistema ultrasonidos + mapa. En las gráficas (a), (b) y (c), para cada periodo de muestreo de la señal de control, se representa la suerte sufrida por las medidas (V válida, NV no válida, MT llega fuera de la ventana de tiempo del filtro, NM no mide, T&NV tarde y no válida, T&V tarde y válida). En las gráficas (d), (e) y (f) se representa si no hay una medida procesable (NMide), y si la hay pero el sistema ultrasonidos + mapa no es capaz de asociarla a ningún objeto (NRecon), o la asocia a una Pared (Pared), o esquina o rincón (Esq/Rin).

C5.2. Resultados obtenidos por el algoritmo IFAsynMixto en la arquitectura robótica propuesta

- Las dos medidas del sensor de ultrasonidos 5 corresponden al obstáculo rectangular, una a su parte delantera (según se aproxima inicialmente el robot) y la otra a su parte posterior. Por lo tanto, no son reconocibles por el algoritmo de asociación, ni asimiladas por el algoritmo IFAsynMixto.
- El primer grupo de medidas del ultrasonidos 0 corresponden a la parte frontal del obstáculo rectangular, y por lo tanto no son asociables a ningún objeto del mapa de la gráfica (a) de la figura C5-39 ni asimilables por el algoritmo IFAsynMixto. El segundo grupo de medidas son producidas por las paredes del entrante inferior del hall. A pesar de corresponder a una zona contemplada en el mapa inicial, el algoritmo de asociación no es capaz de asignarlas a ningún objeto, ya que estas medidas han sido producidas por el rebote de los ultrasonidos en varias paredes.

La existencia de rebotes es apreciable en el mapa de ocupación de la gráfica (b) de la figura C5-40, donde se observa que la probabilidad de ocupación de la parte exterior de la esquina inferior del entrante ha crecido, como consecuencia de la existencia de medidas a una distancia superior a la de los objetos del entorno.

- El mayor número de medidas válidas se corresponden al sensor de ultrasonidos 6, obtenidas cuando el robot se mueve paralelo al eje de las x y próximo a las paredes de la zona inferior. Este hecho se produce durante la primera parte de la trayectoria y al final de la misma (de ahí que haya un conjunto inicial y final de medidas válidas). Las dos medidas realizadas en la parte intermedia de la trayectoria se corresponden a la parte posterior del obstáculo, y por lo tanto son rechazadas. En los subconjuntos de medidas inicial y final no todas las medidas son asignables ni todas las asignables válidas.
- Por último, de las medidas del sensor de ultrasonidos 3, solo hay una válida, que ha sido realizada cuando el robot estaba orientado perpendicularmente al hueco inferior del hall. Las medidas anteriores corresponden a obstáculos (el primer grupo al rectangular y el segundo al obstáculo cilíndrico) y las posteriores al hueco del hall (y no son reconocibles por el algoritmo de asociación por corresponder a medidas rebotadas en los objetos de la esquina inferior del entrante del hall).

Así pues, de todas las medidas realizadas por los sensores de ultrasonidos únicamente son válidas dos proporcionadas por el sensor 4 entre los 90 y 100 segundos (correspondientes a una esquina y pared del hueco inferior del hall), una medida del sensor 3 realizada a los 90 segundos (correspondiente a una pared del mismo hueco), y varias medidas del sensor 6, realizadas antes de los 25 segundos y después de los 99 segundos (todas ellas correspondientes a paredes de la parte inferior del hall, salvo una, realizada en el instante 110, que se corresponde a una esquina del hueco inferior). Además, es importante resaltar que una buena parte de las medidas de ultrasonidos consideradas no válidas para el algoritmo IFAsynMixto corresponden a objetos no presentes en el mapa utilizado por el algoritmo de asociación. Por lo tanto, no son asimiladas por el mismo para

estimar la posición del robot, aunque si son utilizadas por el algoritmo del módulo *Entorno* para actualizar el mapa de probabilidad de ocupación, y de esta forma poder esquivar los obstáculos no recogidos en los mapas probabilísticos iniciales.

De la asimilación de las medidas válidas de la brújula, los encoders, y ultrasonidos, ordenadas y/o retardadas, y de los modelos de movimiento y de los sensores del robot, el algoritmo IFAsynMixto obtiene los resultados recogidos en las gráficas de la figura C5-43.

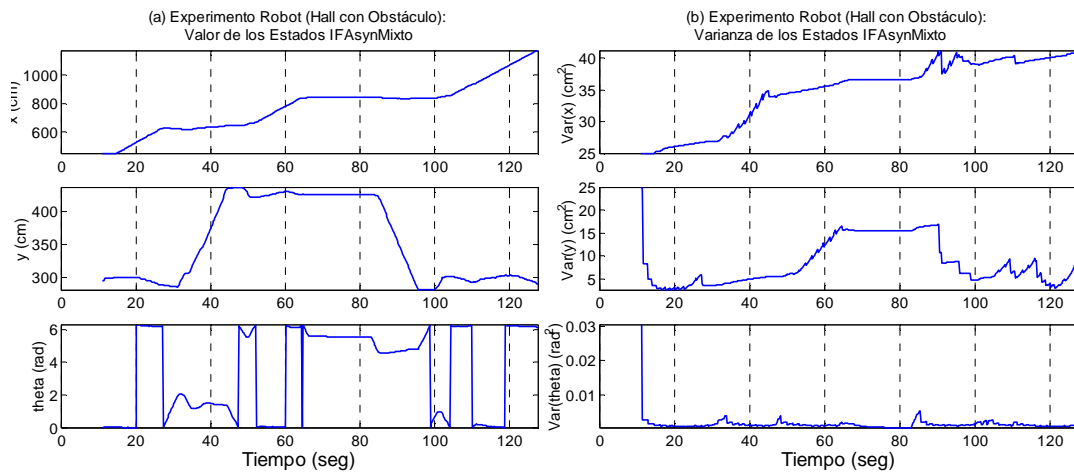


Figura C5-43. Valor estimado de los estados (x , y , θ) y su varianza para el experimento real en el hall con obstáculos.

En las mismas se observa que:

- Hay cambios bruscos de valores en el estado θ , correspondientes a la discontinuidad $0-2\pi$, en torno a la que se encuentra la orientación del robot cuando éste se mueve paralelo al eje x .
- La varianza de los estados no aumenta significativamente entre los 65 y 82 segundos ya que es el tiempo que el robot permaneció parado como medida de seguridad ante la detección de la parte posterior del obstáculo rectangular.
- La varianza de los estados (y , θ) decrece al inicio ya que el elevado valor inicial es corregido por las medidas de los sensores que proporcionan información significativa sobre dichos estados (la brújula y el sensor de ultrasonidos 6). Sin embargo, la varianza del estado x aumenta, ya que la información proporcionada por los encoders no es capaz de reducirla, y la medida de distancia a la pared del sensor de ultrasonidos 6, no contiene información sobre el estado x . El pico de varianza inicial asociada al estado y (a los 25 segundos) se debe a los instantes de muestreo en los que no existen medidas válidas del sensor 6.
- Los picos de varianza del estado θ corresponden a giros del robot y/o a instantes en los que se han rechazado las medidas de dicho sensor.

- La varianza de los estados x e y crece desde el instante 25 hasta el 90, instante en el que se vuelve a disponer de medidas válidas de los ultrasonidos. Los decrementos significativos del estado x corresponden a las esquinas, mientras que los del estado y se deben a la de esquinas y paredes. Este es debido a que las paredes detectadas son paralelas al eje x y por lo tanto, las medidas asociadas a las mimas no contienen información significativa referente al estado x .

Los resultados obtenidos por el algoritmo IFAsynMixto durante los dos experimentos reales responden al comportamiento real del robot, por lo que se comprueba que tanto los modelos utilizados como el algoritmo son adecuados. Es importante resaltar en este punto que el algoritmo IFAsynMixto permite asimilar las medidas desordenadas y retardadas (brújula y ultrasonidos), que de ser desechada por llegar tarde provocarían un crecimiento significativo de la varianza de todos los estados, similar a la observada en el experimento simulado con únicamente medidas de los encoders. Aún más, como la varianza del modelo de movimiento de los experimentos reales es mayor durante las rotaciones del robot, el crecimiento de la varianza de los estados sería superior.

Conclusiones

La investigación realizada en esta tesis se ha centrado en (1) el desarrollo de métodos de fusión para la asimilación secuencial y asíncrona de medidas retardadas y temporalmente desordenadas, y (2) el diseño e implementación de un sistema unificado de integración y fusión multisensorial con medidas temporalmente desordenadas para el control de robots autónomos móviles.

Un objetivo subyacente al desarrollo de los nuevos algoritmos y al diseño e implementación de la arquitectura de fusión y control ha sido la búsqueda de generalidad y versatilidad en ambos casos, de forma que tanto los algoritmos como la arquitectura desarrollados son fácilmente adaptables a diferentes sistemas robóticos. En el caso de los algoritmos, el grado de generalización buscado ha sido todavía mayor, hecho que permite que éstos sean utilizados en diferentes sistemas dinámicos de control, y no sólo sirvan para estimar la posición de un robot autónomo móvil.

El desarrollo de los algoritmos, tarea a la que se ha dado el peso principal en la organización de la memoria, se ha realizado de forma progresiva e incremental. Su origen se encuentra en el modelado del sistema dinámico mediante una red bayesiana y en el uso de un algoritmo de inferencia exacto para estimar el estado del sistema. El algoritmo de inferencia desarrollado bajo estas premisas, denominado Bayes, minimiza el número de operaciones del algoritmo de inferencia exacto original y resuelve el problema del filtrado con una etapa de validación de medidas previa a la asimilación, en los casos en los que las medidas del sistema se encuentren disponibles de forma asíncrona, retardada, y temporalmente desordenada.

La equivalencia existente entre el primer algoritmo desarrollado (Bayes) y el filtro de Información multisensorial, cuando al segundo se le suministran las medidas de forma ordenada y sin retardos, y un cuidadoso estudio de los mecanismos que hacen que el algoritmo Bayes trate de forma automática los retardos y desordenes de las medidas, han llevado al desarrollo del segundo algoritmo propuesto en la tesis bajo el nombre de IFAsyn. Éste es una nueva versión del filtro de Información multisensorial con etapa de validación, que incorpora mecanismos similares al algoritmo Bayes, para tratar de forma automática el desorden y retardos existentes en las medidas del sistema. Aunque tanto Bayes como IFAsyn obtienen los mismos resultados, IFAsyn reduce las necesidades de cómputo y de memoria de Bayes, por lo que resulta más apropiado para estimar secuencialmente el estado de un sistema de control en tiempo real. Aun más, IFAsyn es computacionalmente más estable que Bayes, ya que su implementación reduce los errores de redondeo debido al desorden, mientras que Bayes introduce errores de este tipo.

Estos dos algoritmos eran aplicables a sistemas lineales de control, pero no servían para sistemas no lineales o mixtos, motivo por el que se decidió dar un último paso, consistente en desarrollar una nueva versión del filtro de Información Extendido multisensorial que soporta el asincronismo, los retardos y los desordenes de las medidas. Los cambios en el nuevo algoritmo, denominado IFAsynMixto, van más allá de sustituir en el algoritmo IFAsyn las expresiones del filtro de Información por las expresiones del filtro de Información Extendido, ya que la falta de linealidad en las medidas requiere un tratamiento especial que provoca la reestructuración de las

operaciones realizadas y nuevas necesidades de memoria. Para el caso no lineal, además del algoritmo IFAsynMixto se proponen otras dos versiones del filtro de Información Extendido para medidas asíncronas, retardadas, y temporalmente desordenadas. La primera de las mismas, IFAsynMixtoEsp, es una solución aproximada que reduce las necesidades de memoria y de cómputo del algoritmo IFAsynMixto, pero que no siempre es válida. La segunda, utilizada sobre las pruebas finales del algoritmo cuando éste forma parte de un sistema complejo de control en tiempo real, es una generalización del algoritmo IFAsynMixto más fácil de adaptar a sistemas reales de control.

Las labores de diseño e implementación del sistema unificado de fusión multisensorial de medidas temporalmente desordenadas para robots autónomos móviles se han ido realizando de forma paralela al desarrollo de los algoritmos, con el objeto de poder disponer del banco de pruebas y la arquitectura de fusión y control en la parte final de la tesis. La elección al inicio de la tesis de las redes bayesianas como técnica de fusión ha permitido en todo momento el desarrollo unificado de este sistema de fusión.

El desarrollo de los algoritmos y su implementación en el sistema de localización del robot permiten que este subsistema trate automáticamente el asincronismo, desorden y retardo de las medidas. Para asegurar un comportamiento similar en el subsistema que estima el estado del entorno, se utiliza una nueva red y un nuevo algoritmo de inferencia capaz de tratar de forma automática la información asíncrona, retardada y temporalmente desordenada con el menor número de operaciones posibles.

La versatilidad y portabilidad de la nueva arquitectura de control y fusión desarrollada se sustenta sobre dos hechos. Por una parte, el uso de la herramienta gráfica EdROOM reduce la separación existente entre la etapa de diseño y especificación del sistema, y la de implementación del mismo como un programa de control. Y por otra, una nueva división de la funcionalidad del sistema en un nuevo conjunto de módulos que hace que la sustitución de la funcionalidad de alguno de ellos influya lo menos posible en la funcionalidad del resto del sistema. Estas características permitirán probar diferentes técnicas de planificación, control y fusión; y utilizar la arquitectura sobre diferentes plataformas robóticas, no sólo reales, sino también simuladas.

Finalmente es importante destacar que tanto la arquitectura de control y fusión, como el último de los algoritmos desarrollados, han sido probados con éxito sobre el robot descrito en el apéndice III, que también ha sido diseñado y construido a lo largo de esta tesis.

Resultados y conclusiones

En las siguientes secciones se resumen los principales resultados y conclusiones que se pueden extraer de las labores realizadas durante esta tesis, organizados de acuerdo con los dos objetivos principales propuestos al inicio de la misma: (1) el desarrollo de métodos de fusión para la asimilación secuencial y asíncrona de medidas retardadas y temporalmente desordenadas, y (2) el diseño e implementación de un sistema unificado de integración y fusión multisensorial con medidas temporalmente desordenadas para el control de robots autónomos móviles.

a. *Desarrollo de algoritmos de fusión para la asimilación secuencial y asíncrona de medidas retardadas y temporalmente desordenadas*

A lo largo de esta tesis se han propuesto un conjunto de algoritmos de fusión secuencial para estimar el estado de sistemas dinámicos de control a partir de las medidas proporcionadas asincrónicamente por un conjunto de sensores que pueden ser recibidas en el sistema de fusión temporalmente desordenadas con diferentes retardos:

Los algoritmos Bayes e IFAsyn sirven para estimar secuencial y asincrónicamente el estado de sistemas dinámicos lineales con términos aditivos de ruido gaussiano cuando al sistema de fusión le llegan las medidas retardadas y temporalmente desordenadas, mientras que el algoritmo IFAsynMixto sirve para sistemas dinámicos lineales, no lineales o mixtos.

Se ha demostrado que los algoritmos Bayes e IFAsyn son analíticamente equivalentes al filtro de Información (y por lo tanto, al filtro de Kalman) pero con la salvedad de que a estos últimos, para que den los mismos resultados, hay que entregarle los datos de forma ordenada y sin retardos. El algoritmo IFAsynMixto es analíticamente equivalente al filtro de Información Extendido (o Kalman Extendido) para las ecuaciones no lineales del sistema y al filtro de Información (y Kalman) para las ecuaciones lineales, cuando los filtros utilizan los mismos datos de forma ordenada y sin retardos.

Es importante destacar que el algoritmo óptimo IFAsyn es una solución equivalente a otras soluciones propuestas recientemente en la literatura ([Mallick2001], [Nettleton2001], [ZhangK2002], [Bar-Shalom2002a], [Bar-Shalom2002b], [Challa2003], [ZhangK2003]), aunque es más general. Por una parte, su formulación permite que sea utilizado para sistemas dinámicos lineales cuya matriz de transición de estados es no invertible (cosa que no ocurre con todos los métodos anteriores). Por otra, la formulación de nuestro algoritmo incluye la existencia de la señal de control (que la mayoría no tiene), una etapa opcional de validación previa a la asimilación de las medidas (no contemplada en el resto), y el carácter multisensorial del sistema (que la mayoría no considera). Además, la división que se establece entre la información sensorial de un instante de

muestreo y de los anteriores en el algoritmo IFAsyn reduce las causas de errores de redondeo debido a la asimilación desordenada de las medidas. En definitiva, nuestro algoritmo es más general y completo que el resto de los algoritmos similares recientemente desarrollados, tal como puede verse en la tabla comparativa C3-10.

Sobre el algoritmo IFAsynMixto cabe destacar que es una solución novedosa para el problema de los sistemas no lineales y mixtos. Se ha demostrado su equivalencia con el filtro de Información Extendido. Presenta como inconveniente que por cada sensor no lineal se aumenta el gasto computacional y las necesidades de memoria del algoritmo. Para evitar este problema, se ha desarrollado el algoritmo IFAsynMixtoEsp, que es una solución aproximada (que no cumple totalmente la equivalencia) para sistemas con funciones no lineales suaves pero que reduce el gasto computacional y de memoria asociado a los sensores no lineales. El uso de esta versión adaptada se encuentra justificada, por el hecho de que el filtro de Información Extendido sobre el que se apoyan las operaciones del algoritmo IFAsynMixto constituye en sí mismo una aproximación subóptima a la solución del problema.

Por último, es importante destacar las siguientes características de los tres algoritmos desarrollados:

- *Permiten un modelado del comportamiento del sistema sobre dos redes bayesianas con la misma estructura (grafo dirigido y acíclico) pero con diferentes tipos de relaciones probabilísticas entre sus variables.* En el caso de los algoritmos Bayes e IFAsyn las relaciones existentes entre las variables del sistema son lineales e incluyen un término aditivo de ruido gaussiano, mientras que en el caso del algoritmo IFAsynMixto las relaciones existentes entre las variables puede ser lineales o no lineales e incluyen un término aditivo de ruido gaussiano.
- *Son capaces de obtener en todo momento la “mejor” estima del estado del sistema para los últimos N instantes de muestreo, a partir de todas las medidas realizadas que han sido recibidas hasta el momento por el algoritmo. Con “mejor” estima del estado nos referimos a aquella que utiliza toda la información no asociada a instantes de muestreo posteriores y que se encuentre disponible en cada momento, haya sido recibida con o sin retardos, de forma ordenada o desordenada.* La utilización de una ventana de tiempo de tamaño fijo (N), elegido antes de la inicialización del algoritmo, sólo impone la necesidad de determinar antes de su uso los retardos máximos introducidos por los algoritmos de pre-procesamiento y el sistema de comunicación.
- *Son técnicas para estimar el estado de sistemas dinámicos de control en tiempo real.* Para conseguir este objetivo se han minimizado el número de operaciones y las necesidades de memoria de cada uno de los algoritmos. Los algoritmos de Bayes e IFAsyn son equivalentes entre sí, aunque las diferencias existentes en el algoritmo soporte sobre el que han sido desarrollados hacen que IFAsyn consuma menos memoria y realice menos operaciones.

Aun más, durante el análisis mediante simulación del funcionamiento de los dos algoritmos se ha observado que la estabilidad computacional del algoritmo IFAsyn es mayor que la del algoritmo de Bayes. A pesar de que *las necesidades de cómputo y de memoria del algoritmo IFAsynMixto crecen a la vez que el número de sensores no lineales*, este algoritmo sigue siendo utilizable dentro de un sistema de control, tal y como se comprueba en los experimentos realizados sobre el robot real (donde 9 de los 10 sensores reciben un tratamiento no lineal), siempre y cuando los cálculos asociados a las medidas no lineales no sobrecarguen en exceso a la totalidad del sistema de control.

- *Incorporan una etapa de validación previa a la asimilación de las medidas, para que los algoritmos sean robustos ante fallos de los sensores.* En las experimentos simulados se ha comprobado como la combinación de medidas desordenadas y una etapa de validación (en los algoritmos propuestos el test chi-cuadrado) que depende del valor del estado da lugar a que los resultados obtenidos en algunos instantes sean ligeramente diferentes a los que obtendrían los mismos algoritmos (o sus equivalentes, los filtros de Información y Kalman) con los mismos datos ordenados. Sin embargo, consideramos que la etapa de validación es fundamental en algunas ocasiones para rechazar las medidas cuyos errores no se encuentran modelados en la matriz de covarianza, como ocurre en los experimentos con nuestro robot en los que las medidas de la brújula están fuertemente afectadas por las fluctuaciones del campo magnético dentro del edificio y hay medidas erróneas de distancia del sistema ultrasonidos + mapa, causadas por los rebotes de las señales de ultrasonido en los rincones.
- *Son fáciles de configurar y adaptar a diferente número y tipo de sensores, así como a diferentes sistemas dinámicos de control*, hecho que se ha puesto de manifiesto durante las fases de análisis de los resultados simulados y las pruebas de los algoritmos dentro de la arquitectura de control. La configuración de los algoritmos se realiza a través de la definición de las ecuaciones que modelan las relaciones entre transiciones de estado, y entre las medidas y el valor de los estados; el tipo de modelo de medida utilizado para cada sensor (lineal o no lineal); las matrices de covarianza asociadas al término aditivo de ruido de dichas expresiones; el tamaño de la ventana temporal; etc. *Esta facilidad de configuración teórica ha sido transferida a las implementaciones del algoritmo en C++ y Matlab*, para facilitar las labores de análisis del funcionamiento de los algoritmos y de reutilización de los mismos sobre diferentes sistemas dinámicos.

b. *Diseño e implementación de un sistema unificado de integración y fusión multisensorial de medidas temporalmente desordenadas para robots autónomos móviles*

La nueva arquitectura de fusión y control ha sido probada con éxito sobre el robot diseñado y construido, a lo largo de este trabajo de investigación. Para ello, se ha diseñado un sistema de fusión unificado, mediante el uso de una sola metodología, las redes bayesianas dinámicas, en los dos subsistemas en los que se divide la funcionalidad del sistema de fusión: uno encargado de estimar la posición del robot y otro de estimar el estado del entorno. Esta unificación metodológica permitirá unir en el futuro, a través de la inclusión de nuevos nodos y enlaces en la red, los dos subsistemas de fusión, de forma que se establezca un flujo de información probabilístico entre los dos subsistemas ahora existentes, y/o aumentar la funcionalidad de todo el sistema de fusión.

Las características del nuevo sistema de fusión de la arquitectura de control son las siguientes:

- *Realiza de forma automática la asimilación secuencial de medidas asíncronas, desordenadas y temporalmente retardadas, tanto en el módulo de localización como en el módulo que construye un mapa probabilístico del entorno. Para lograr este objetivo ha sido necesario implementar en el módulo que distribuye la información entre los dos subsistemas de fusión multisensorial mecanismos que aseguran que el intercambio de información entre ellos tiene en cuenta los posibles desordenes y retardos de las medidas.*
- *Utiliza el algoritmo IFAsynMixto en el subsistema que estima la posición del robot a partir de toda la información de localización recibida, de forma que la posición estimada sea independiente del orden y retardos en la recepción de las medidas. Durante la implementación del algoritmo dentro de la arquitectura de fusión y control se ha asegurado la adaptabilidad del subsistema de localización a diferentes tipos y número de sensores, y/o diferentes robots, asociando la funcionalidad de este módulo a una librería externa, modificable en cualquier momento.*
- *Utiliza una nueva red bayesiana y un nuevo algoritmo de inferencia para estimar el mapa probabilístico de ocupación del entorno, algoritmo en el que se han minimizado el número de mensajes intercambiados entre los módulos del subsistema y que sustenta la fusión asíncrona de medidas desordenadas y retardadas. La funcionalidad de sus módulos también se define en un conjunto de librerías externas, para facilitar las labores de sintonización del sistema a los sensores disponibles en la plataforma robótica sobre la que se utiliza la arquitectura de control y fusión.*

Por último, también conviene destacar que la totalidad de la arquitectura de control y fusión se caracteriza por una elevada versatilidad y portabilidad, lo que permite probar dentro de la misma diferentes algoritmos de planificación, control y fusión; y utilizar la arquitectura sobre diferentes

tipos y número de sensores y/o plataformas sensoriales autónomas móviles. Estas características se han conseguido con una cuidadosa división de la funcionalidad en módulos especialmente orientada al desarrollo y prueba de algoritmos de fusión, y por medio del uso de la herramienta EdROOM que permite especificar gráficamente la estructura y el comportamiento de todo el sistema, y que genera automáticamente el código C++ asociado a la definición gráfica. De esta forma, un cambio en el diseño se traduce, mediante el generador de código, en un cambio en el programa, hecho que asegura la consistencia entre la especificación del sistema y el funcionamiento del programa. Además, el uso de esta herramienta permite utilizar el mismo código en diferentes sistemas operativos, motivo que incrementa la portabilidad de la arquitectura de control diseñada. La portabilidad del código se encuentra vinculado a la existencia de una librería de servicios asociada al sistema operativo, y ante la falta de una librería para el sistema operativo Windows, sobre el que se han realizado las pruebas de la arquitectura sobre el robot real y que permite el funcionamiento del sistema bajo condiciones de tiempo real laxas, dicha librería también ha sido desarrollada como una tarea auxiliar de la tesis.

Líneas de trabajo futuro

La versatilidad de la arquitectura de fusión y control utilizada permitirá probar sobre la misma diferentes estrategias de planificación, control y fusión; o incrementar la funcionalidad del sistema a través de la incorporación de nuevos módulos. Además, la versatilidad del sistema de fusión permitirá que éste sea reutilizado, tras las modificaciones necesarias, para el control de otras plataformas autónomas móviles que dispongan de un sistema sensorial que proporcionen información sobre la existencia de objetos en el entorno y respecto a la posición del robot.

Entre las posibles líneas de trabajo futuro, sustentadas por la versatilidad de los algoritmos y sistemas desarrollados, es importante destacar nuestro interés por:

- Unificar el sistema de fusión no sólo bajo el uso de una sola metodología en los dos subsistemas, si no a través de la construcción de una única red bayesiana (en vez de las dos que existen en la actualidad) que permita el flujo probabilístico de la incertidumbre y conocimiento entre los dos subsistemas ahora existentes.
- Aumentar la funcionalidad del sistema de fusión mediante el uso de redes bayesianas de mayor complejidad y de diferentes algoritmos que, además de inferir el estado de distintas variables, permitan la toma de decisiones, determinar qué información es más fiable o significativa en el proceso de inferencia, etc.
- Explotar las múltiples posibilidades del sistema de visión dentro del sistema de fusión, por el medio del uso de redes bayesianas que permitan fusionar la gran cantidad de información extraíble, con el uso de diferentes algoritmos, de las secuencias de imágenes tomadas por

dos cámaras (visión estereoscópica y visión de movimiento), tanto para determinar la posición del robot como para conocer la situación del entorno.

- Estudiar la posibilidad de realizar un uso combinado de los algoritmos implementados, junto con otros, como los filtros de Partículas, con el objeto de resolver otros problemas de localización del robot como son la determinación de su posición inicial cuando ésta es desconocida o cuando la incertidumbre existente sobre la posición del robot es tan elevada que el robot se encuentra perdido.
- Probar y desarrollar estrategias de control que tengan en cuenta la incertidumbre existente sobre la posición del robot y sobre el estado del entorno.

Publicaciones realizadas

A continuación se enumeran los trabajos y publicaciones realizados durante el desarrollo de esta tesis:

- G. Pajares, J. M. de la Cruz, E. Besada-Portas, J.A. López-Orozco. *Selección de Propiedades en la Correspondencia Estereoscópica para Sistemas de Visión y Robots Autónomos*. Hispabot 2003. Madrid. 2003.
- G. Pajares, E. Besada-Portas, J.M de la Cruz. *Analysis of Support Vector Machines and Bayesian Methods for Sunflower Classification*. Recent Research Development in Pattern Recognition. Vol 3. Transworld Research Network.2002.
- E. Besada-Portas, O. Rodriguez-Polo, J.A. López-Orozco, J.M. de la Cruz. *Versatile Sensing Architecture for Autonomous Mobile Robots*. In the Proceedings of the Sixth World Multiconference on Systematics, Cybernetics and Informatics. Orlando, Florida, USA. 2002.
- E. Besada-Portas, J.A. Lopez-Orozco, J.M. de la Cruz. *Unified Fusion System Based on Bayesian Networks for Autonomous Mobile Robots*. Proceedings of the 5th International Conference in Information Fusion. Annapolis, Maryland, USA. 2002.
- E. Besada-Portas. *Extension and Modification of the G&T System. Bayesian Approach to Data Analysis*. Technical Report CDS-TR-00-11. Department of Computer Science. Royal Holloway. University of London. Reino Unido. 2000.
- E. Besada-Portas. *Modified Xpress- Probabilistic Reasoning Expert System Under X-Windows*. Technical Report CDS-TR-00-10. Department of Computer Science. Royal Holloway. University of London. Reino Unido. 2000.

- J.A. Lopez-Orozco, J.M. de la Cruz, E. Besada & P. Rupierez. *An Asynchronous, Robust and Distributed Multisensor Fusion System for Mobile Robots*. The International Journal of Robotics Research. Vol 19, No 10. 2000.
- E. Besada-Portas. *Utilización de Visión Artificial en la Integración Multisensorial de Robots Móviles Autónomos*. Trabajo de Doctorado. UCM. 1999
- O. Rodríguez-Polo, J.M. de la Cruz, J.A. López-Orozco. E. Besada. *Diseño de Prácticas de Control Digital Utilizando Multi-Tarea en Tiempo Real y Modelado Orientado a Objetos*. XIX Jornadas de Automática. Madrid. 1998.
- J.A. Lopez-Orozco, J. M. de la Cruz, E. Domínguez, E. Besada, O. R. Polo. *An Open Sensing Architecture to Autonomous Mobile Robots*. Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation. ISCI/CIRA/ISAS Joint Conference. Gaithersburg, Maryland, USA. 1998.

Bibliografía

- Abidi1992 M.A. Abidi, R.C. Gonzalez. *Data Fusion in Robotics and Matching Intelligence*. Academic Press. 1992.
- Alami1998 R. Alami, R. Chatila, S. Fleury, M. Ghallab, F. Ingrand. *An Architecture for Autonomy*. International Journal of Robotics Research. Vol. 17. No 4. 1998.
- Albus1987 J.S. Albus, R. Lumia, H. McClain. *NASA/NBRS standard reference model for telerobot control system architecture (NASREM)*. Technical Report. 1987.
- Albus1995 J.S. Albus, C.R. Malean, A. J. Barbera, M.L. Fitzgerald. *Hierarchical Control for Robots and Teleoperators*. Proc. of the IEEE Workshop on Intelligent Control. 1985.
- Almond1995 R. Almond. *Graphical Models*. Chapman & Hall. 1995.
- Anderson1979 B.D.O. Anderson, J.B. Moore. *Optimal Filtering*. Prentice-Hall. 1979.
- Andreassen1987 S.A. Andreassen, M. Woldbye, B. Falck, S.K. Andersen. *MUNIN. A Causal Probabilistic Network for Interpretation of Electromyographic Findings*. Proc. of the 10th Joint Conference on Artificial Intelligence. 1987.
- Arkin1989 R.C. Arkin. *Motor Schema-Based Mobile Robot Navigation*. The International Journal of Robotics Research. Vol. 8, No 4. 1989.
- Arkin1992 R.C. Arkin. *A Hybrid Reactive/Hierarchical Robot Architecture*. In Workshop on Architecture for Intelligent Control System. IEEE International Conference on Robotics and Automation. 1992.
- Arkin1998 R.C. Arkin. *Behaviour-Based Robotics*. The MIT Press. 1998.
- Bangsø2000 O. Bangsø, P.H. Willemin. *Top-Down Construction and Repetitive Structures Representation in Bayesian Networks*. Proc. of the 13th International Florida Artificial Intelligence Research Symposium Conference. AAAI Press. 2000.

Bibliografía

- Barber2001 R. Barber, M.A. Salichs. *A new Human Based Architecture for Intelligent Autonomous Robots*. In the 4th IFAC Symposium on Intelligent Autonomous Vehicles. 2001.
- Bar-Shalom1993 Y. Bar-Shalom, X.R. Li. *Estimation and Tracking: Principles, Techniques and Software*. Artech House. 1993.
- Bar-Shalom2001 Y. Bar-Shalom, X.R. Li, T. Kirubarajan. *Estimation with Applications to Tracking Navigation*. Wiley-Interscience 2001.
- Bar-Shalom2002a Y. Bar-Shalom. *Update with Out-of-Sequence Measurements in Tracking: Exact Solution*. IEEE Transactions on Aerospace and Electronic Systems, Vol. 38, no 3. 2002.
- Bar-Shalom2002b Y. Bar-Shalom, M. Mallick, H. Chen, R. Washburn. *One Step Solution for the General Out of Sequence Measurement Problem in Tracking*. IEEE Aerospace Conference. 2002.
- Becker1996 A. Becker, D. Geiger. *A sufficiently Fast Algorithm for Finding Close to Optimal Junction Trees*. Proc. of the 12th Annual Conference on UAI. 1996.
- Bennet1994 S. Bennet. *Real Time Computer Control. An Introduction*. Ed. Prentice-Hall. 1994.
- Berler1997 A. Berler, S.E. Shimony. *Bayes Networks for Sonar Sensor Fusion*. Proc. of the 13th Annual Conference on UAI. 1997.
- Besada-Portas1999 E. Besada-Portas. *Utilización de Visión Artificial en la Integración Multisensorial de Robots Móviles Autónomos*. Trabajo de Doctorado. UCM. 1999.
- Besada-Portas2000a E. Besada-Portas. *Extension and Modification of the G&T System. Bayesian Approach to Data Analysis*. Technical Report CDS-TR-00-11. Department of Computer Science. Royal Holloway. University of London. 2000.
- Besada-Portas2000b E. Besada-Portas. *Modified Xpress- Probabilistic Reasoning Expert System Under X-Windows*. Technical Report CDS-TR-00-10. Department of Computer Science. Royal Holloway. University of London. 2000.
- Besada-Portas2002a E. Besada-Portas, J.A. Lopez-Orozco, J.M. de la Cruz. *Unified Fusion System Based on Bayesian Networks for Autonomous Mobile Robots*. Proc. of the 5th International Conference in Information Fusion. 2002.

-
- Besada-Portas2002b E. Besada-Portas, O. Rodriguez-Polo, J.A. López-Orozco, J.M. de la Cruz. *Versatile Sensing Architecture for Autonomous Mobile Robots*. In the Proceedings of the 6th World Multiconference on Systematics, Cybernetics and Informatics. 2002.
- Betta2000 G. Betta, A. Pietrosanto. *Instrument Fault Detection and Isolation: State of the Art and New Research Trends*. IEEE Transactions on Instrumentation and Measurement. Vol. 49. 2000.
- Bishop1995 C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press. 1995.
- Bonasso1997 R.P. Bonasso, D. Kortenkamp, D.P. Millar, M.Slack. *Experiences with an Architecture for Intelligent Reactive Agents*. Journal of Experimental Theory of Artificial Intelligence. 1997.
- Borenstein1991 J. Borenstein, Y. Koren. *Histogram in-Motion Mapping for Mobile Robot Obstacle Avoidance*. IEEE Journal of Robotics and Automation. Vol. 7. No 4. 1991.
- Bouchon-Meunier1999 B. Bouchon-Meunier, R.R. Yager, L.A. Zadeh. *Information, Uncertainty and Fusion*. Kluwer International Series in Engineering and Computer Science. Springer. 1999.
- Bouckaert1996 R.R. Bouckaert, E. Castillo, J.M. Gutierrez. *Modified Simulation Scheme for Inference in Bayesian Networks*. International Journal of Approximate Reasoning. 1996.
- Breese1996 J.S. Breese, D. Heckerman. *Decision-Theoretic Troubleshooting: A Framework for Repair and Experiment*. Proc. of the 12th Conference on UAI. 1996.
- Brooks1986 R. A. Brooks. *A Robust Layered Control System for a Mobile Robot*. IEEE Journal of Robotics and Automation. Vol. RA-2, No 1. 1986.
- Brooks1991 R. A. Brooks. *Elephant's don't play chess*. In Designing Autonomous Agents. 1991.
- Brooks1998 R.R. Brooks, S.S. Iyengar. *Multi-Sensor Fusion*. Fundamentals and Applications with Software. 1998
- Buchanan1984 B.G. Buchanan, E.H. Shortliffe. *Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley. 1984
- Buntine1994 W.L. Buntine. *Operations for Learning with Graphical Models*. Journal of Artificial Intelligent Research. Vol. 2. 1994.

Bibliografía

- Calvache2001 M.M. Calvache Sánchez. *Estructura 3D mediante visión estereoscópica*. Proyecto fin de carrera de Ingeniería Electrónica, Departamento de Arquitectura y Automática, UCM. 2001
- Campion1996 G. Campion, G. Bastin, B. D' Andrea-Novel. *Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots*. IEEE Transactions on Robotics and Automation. Vol. 12, No 1. 1996.
- Campos1999 L.M. Campos, J.A. Gámez, S. Moral. *Partial Abductive Inference in Bayesian Belief Networks: An Evolutionary Computation Approach by using Problem Specific Genetic Operators*. Pattern Recognition Letters. 1999.
- Campos2000 L.M. Campos, J.F. Huete. *Approximating Causal Orderings for Bayesian Networks Using Genetic Algorithms and Simulated Annealing*. Proc. of the 8th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU). 2000.
- Castellanos2001 J.A. Castellanos, J. Neira, J.D. Tardós. *Multisensor Fusion for Simultaneous Localization and Map Building*. IEEE Transactions on Robotics and Automation, Vol. 17, No 6. 2001.
- Castillo1997 E. Castillo, J.M. Gutiérrez, and A.S. Hadi. *Expert Systems and Probabilistic Network*. Monografías de la Academia Española de Ingeniería, Madrid. 1997.
- Challa2003 S. Challa, R. Evans, X. Wang. *A Bayesian Solution and its Approximations to Out-of-Sequence Measurement Problems*. Information Fusion 4. Elsevier, 2003.
- Chatilla1995 R.G. Chatilla. *Deliberation and Reactivity in Autonomous Mobile Robots*. Robotics and Autonomous Systems. 1995.
- Cherkassky1998 V. Cherkassky, F. Mulier. *Learning from Data: Concepts, Theory and Methods*. Wiley. 1998.
- Cooper1990 G.F. Cooper. *The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks*. Artificial Intelligence. Vol. 42. 1990.
- Cowell1992 R. Cowell, A. Dawid. *Fast Retraction of Evidence in a Probabilistic Expert System*. Statistics and Computing. 1992.
- Cowell1994 R.G. Cowell. *Decision Networks A New Formulation for Multistage Decision Problems*. Research Report 109. Department of Statistical Science, University College London. 1994.

-
- Cowell1998 R. Cowel. *Advanced Inference in Bayesian Networks*. Learning in Graphical Models. Ed. by M. Jordan. The MIT Press. 1998.
- Cowell1999 R.G. Cowell, A.P. Dawid, S.L. Lauritzen, D.J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag New York Inc. 1999.
- Cox1991 I. Cox. *Blanche- An Experiment in Guidance and Navigation of An Autonomous Robot Vehicle*. IEEE Transactions on Robotics and Automation. Vol. 7, No 2. 1991.
- Dagum1993 P. Dagum, M. Luby. *Approximating Probabilistic Inference in Bayesian Networks is NP-Hard*. Artificial Intelligence, Vol. 60. 1993.
- Dawid1995 A.P. Dawid, U. Kjærulff, S.L. Lauritzen. *Hybrid propagation in junction trees*. In: Advances in Intelligent Computing. Ed. by B. Bouchon, R.R. Yager, L.A. Zadeh. Springer Verlag. Berlin. 1995.
- Dechter1996 R. Dechter. *Bucket elimination: A unifying framework for probabilistic inference*. Proc. of the 12th Annual Conference on UAI. 1996.
- Dellaert1999 F. Dellaert, W. Burgard, D. Fox, S. Thrun. *Using the Condensation Algorithm for Robust, Vision-based Mobile Robot Localization*. Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1999.
- Dempster1967 A.P. Dempster. *Upper and Lower Probabilities Induced by a Multivalued Mapping*. Annals of Mathematical Statistics, Vol. 38, No 2. 1967.
- Dickison1997 S.J. Dickinson, H.I. Christensen, J.K. Tsotsos, G. Olofsson. *Active Object Integrating Attention and Viewpoint Control*. Computer Vision and Image Understanding, 67(3). 1997.
- Diez1994a J. Diez. *Sistema Experto Bayesiano para Ecocardiografía*. Tesis Doctoral. UNED. 1994.
- Diez1994b F.J. Diez, J. Mira. *Distributed inference in Bayesian Networks*. Cybernetic and Systems, 25. 1994.
- Diez1996 F.J. Diez. *Local Conditioning in Bayesian Networks*. Artificial Intelligence. 1996.
- Doucet2000 A. Doucet, N. Freitas, K. Murphy, S. Russell. *Rao-Blackwellised particle filtering for dynamic Bayesian networks*. Proc. of the 16th Annual Conference on UAI. 2000.

Bibliografia

- Doucet2001 A. Doucet, N. Freitas, N. Gordon (Eds). *Sequential Monte Carlo Methods in Practice*. Springer. 2001.
- Duda1979 R.O. Duda, P.E. Hart, K. Konolige, R. Reboh. *A Computer-Based Consultant for Mineral Exploration*. Technical Report. SRI International. 1979.
- Durrant-Whyte1988 H.F. Durrant-Whyte. *Integration, Coordination and Control of Multi-sensor Robot Systems*. Kluwer Academic Publisher. 1988.
- Elfes1986 A. Elfes. *A Distributed Control Architecture for an Autonomous Mobile Robot*. Artificial Intelligence, Vol. 1, No 2. 1986.
- Elfes1990 A. Elfes. *Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception*. Proc. of the 6th Annual Conference on UAI. 1990.
- Everett1995 H.R. Everett. *Sensors for Mobile Robots. Theory and Application*. A. K. Peters. Wellesley. 1995.
- Firby1994 R.J. Firby. *Task Networks for Controlling Continuous Processes*. In Sensorial International Conference on AI Planning Systems. 1994.
- Fleury1994 S. Fleury, M. Herrb, R. Chatila. *Design of a modular Architecture for Autonomous Robot*. Proc. IEEE International Conference on Robotics and Automation. 1994.
- Fox2001 D. Fox, S. Thrun, W. Burgard, F. Dellaert. *Particle filters for mobile robot localization*. In Sequential Monte Carlo Methods in Practice, Ed. by A. Doucet, N. Freitas, N. Gordon. Springer. 2001.
- Friedman1998 N. Friedman, K. Murphy, S. Russell. *Learning the Structure of Dynamic Bayesian Networks*. Proc. of the 14th Annual Conference on UAI. 1998.
- Fukunaga1993 K. Fukunaga. *Statistical Pattern Recognition*. Handbook of Pattern Recognition and Computer Vision. Eds. C.H. Chen, L.F. Pau, P.S.P. Wang. World Scientific Publishing Company. 1993.
- Fung1990 R. Fung, K. Chang. *Weighting and Integrating Evidence for Stochastic Simulation in Bayesian Networks*. In Uncertainty in Artificial Intelligence 5. Ed. by M. Henrion, R.D. Shachter, L.N. Kanal, J.F. Lemmer. 1990.
- Fusion2004 *Fusion'2004. The 7th International Conference in Information Fusion*. Proc. of the IEEE. International Society of Information Fusion (<http://www.inforfusion.org>).

-
- Gaag1999 L.C. van der Gaag, U. Kjærulff. *Making Sensitivity Analysis Computationally Efficient*. Proc. of the 16th Annual Conference on UAI. 1999.
- Gachet1994 D. Gachet, M.A. Salichs, L. Moreno, J.R. Pimentel. *Learning Emergent Tasks for an Autonomous Mobile Robot*. Proc. of the IEEE/RSJ/GI. International Conference on Intelligent Robots and Systems. Advanced Robotic System and the Real World. 1994.
- Gamerman1994 A. Gamerman. *Probabilistic Reasoning and Bayesian Belief Networks*. Alfred Waller, Henley-on-Thames. 1994.
- Gamerman1996 A. Gamerman. *Computational Learning and Probabilistic Reasoning*. John Wiley & Sons. 1996
- Gat1992 E. Gat. *Integrating planning and reacting in a heterogeneous asynchronous architecture for Controlling Real-world Mobile Robots*. In Tenth National Conference on Artificial Intelligence. AAAI. 1992.
- Gelb1974 A. Gelb. *Applied Optimal Estimation*. The MIT Press. 1974
- Gordon1993 N.J. Gordon, D. J. Salmond, A.F.M. Smith. *Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation*. IEE Proceedings-F. Vol. 140, No 2. 1993.
- Hager1990 G. D. Hager. *Task-Directed Sensor Fusion and Planning. A Computational Approach*. Kluwer. 1990.
- Hall1992 D.L. Hall. *Mathematical Techniques in Multisensor Data Fusion*. Artech House. Boston. 1992.
- Hall1997 D.L. Hall, J. Llinas. *An Introduction to Multisensor Data Fusion*. Proc. of the IEEE, Vol. 85, No. 1, pp. 6-23, 1997.
- Hall2001 D.L. Hall, J. Llinas. *Handbook of Multisensor Data Fusion*. CRC Press. 2001.
- Hall2004 D.L. Hall, S.A.H. McMullen. *Mathematical Techniques in Multisensor Data Fusion*. Artech House Information Warfare Library. 2004.
- Harel1987 D. Harel. *Statecharts: A Visual Formalism for Complex Systems*. Science of Computer Programming 8. 1987.
- Haykin1999 S. Haykin. *Neural Networks: A Comprehensive Foundation*. 2nd Edition. Prentice Hall 1999.
- Hearst1997 M.A. Hearst, D. Draper, D. Madigan, L. Breiman, W. Buntine. *Banter on Bayes: Debating the Usefulness of Bayesian Approaches to Solving Practical Problems*. IEEE Expert: Intelligent Systems. 1997.

Bibliografía

- Heckerman1986 D. Heckerman. *Probabilistic Interpretation for MYCIN's certainty factors*. Uncertainty in Artificial Intelligence. Ed. by L.N. Kanal, J.F. Lemms. 1986.
- Heckerman1995 D. Heckerman, A. Mamdani, M.P. Wellman. *Real-World Applications of Bayesian Networks*. Communications of the ACM, Vol. 38, No 3. 1995.
- Heckerman1996 D. Heckerman. *A Tutorial on Learning with Bayesian Networks*. Microsoft Research Technical Report. MSR-TR-95-06. 1996.
- Henrion1988 M. Henrion. *Propagation of Uncertainty in Logic Sampling in Bayes' Networks*. Uncertainty in Artificial Intelligence. 1988.
- Henrion1991 M. Henrion. *Search-Based Methods to Bound Diagnostic Probabilities in Very Large Belief Nets*. Proc. of the 7th Conference on UAI. 1991.
- Hernandez1997 L.D. Hernández, S. Moral. *Mixing Exact and Importance Sampling Propagation Algorithms in Dependence Graphs*. International Journal of Intelligent Systems. 1997.
- Herranz-Molinero2003 M. Herranz-Molinero, O. Rodríguez-Polo, L. Piñuel, J.M. de la Cruz. *EdROOM. A CASE Tool for ROOM Based Modelling and Automatic Code Generation of Real Time Systems Running Under RTAI*. 5th Real Time Linux Workshop, Spain. 2003.
- Hilton1993 R.D. Hilton, D.A. Martin, W.D. Blair. *Tracking with Time Delayed Data in Multisensor systems*. NSVCDD/TR-93/351, Dahlgren, VA. 1993.
- Houzelle1994 S. Houzelle, G. Giraudon. *Contribution to Multisensor Fusion Formalization*. Robotics and Autonomous Systems. Vol. 13. 1994.
- Howard1984 R.A. Howard, J.E. Matheson. *Influence Diagrams*. In R.A. Howard and J.E. Matheson Eds, Readings on the Principles and Applications of Decision Analysis. Vol. 2. 1984.
- I2C_Specs *The I2C bus specification. Version 2.1*. Document order number 939839349911. Phillips Semiconductors.
- Ibargüengoytia1996 P.H. Ibargüengoytia, L.E. Sucar, S. Vadera. *A Probabilistic Model for Sensor Validation*. Proc. of the 12th Annual Conference on UAI. 1996.
- Ibargüengoytia1998 P. Ibargüengoytia, L. Sucar, S. Vadera. *Any Time Probabilistic Reasoning for Sensor Validation*. Proc. of the 14th Annual Conference on UAI. 1998.
- InformationFusion B.V. Dasarathy (Editor-in-chief). *Information Fusion. An International Journal on Multi-Sensor, Multi-Source Information Fusion*. Elsevier (www.elsevier.com).

-
- IPMU2004 *Proc. of the Information Processing and Management of Uncertainty in Knowledge Based Systems*. <http://ipmu2004.dipmat.unipg.it> . 2004.
- Isard1996 M. Isard, A. Blake. *Contour Tracking by Stochastic Propagation of Conditional Density*. European Conference on Computer Vision. 1996.
- ISSNIP'04 *The International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. 2004.
- Jackings1983 C.L. Jackings, S.L. Tanimoto. *Quad-Trees, Oct-Trees and K-Trees: A Generalized Approach to Recursive Decomposition of Euclidean Space*. IEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 5, No 5. 1983.
- JDL1991 *Data Fusion Lexicon*. U.S. Department of Defense, Data Fusion Subpanel of the Joint Directors of Laboratories. 1991.
- JensenCS1995 C.S Jensen, A. Kong, U. Kjærulff. *Blocking Gibbs sampling in very large probabilistic expert systems*. International Journal of Human-Computer Studies, 42. 1995.
- JensenCS1998 K.S. Jensen. *A Simple Method for Finding a Legal Configuration in Complex Bayesian Networks*. Statistics and Computing. 1998.
- JensenFV1990a F.V. Jensen, K.G. Olesen, S.K. Andersen. *An Algebra of Bayesian Belief Universe for Knowledge Systems*. Networks, Vol. 20. No. 5. 1990.
- JensenFV1990b F.V. Jensen, S. L. Lauritzen, K.G. Olesen. *Bayesian Updating in causal probabilistic networks by local computation*. Computational Statistics Quarterly 4. 1990.
- JensenFV1990c F.V. Jensen, J. Nielsen, H.I. Christensen. *Use of Probabilistic Networks as high level models in Computer Vision*. Technical Report 90-39, Department of Mathematics and Computer Science, University of Aalborg. 1990.
- JensenFV1994a F.V. Jensen, J. Liang. *drHugin: A system for value of Information in Bayesian Networks*. Proc. of the 5th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU). 1994.
- JensenFV1994b F.V. Jensen, F. Jensen. *Optimal Junction Trees*. Proc. of the 10th Annual Conference on UAI. 1994.
- JensenFV1995 F.V. Jensen. *Cautious Propagation in Bayesian Networks*. Proc. of the 11th Annual Conference on UAI. 1995.

Bibliografia

- JensenFV1996 F.V. Jensen. *An introduction to Bayesian Networks*. Springer-Verlag New York Inc. 1996.
- JensenFV1999 F.V. Jensen, S.L. Lauritzen. *Probabilistic Networks*. www.citeseer.ist.psu.edu/jensen97probabilistic.html. 1999.
- JensenFV2001a F.V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag. 2001.
- JensenFV2001b F.V. Jensen. *Influence diagrams*. Encyclopaedia of Environmetrics. John Wiley & Sons. 2001
- Jian-Ming1994 L. Jian-Ming, H.I. Christensen, F.V. Jensen. *Qualitative Recognition Using Bayesian Reasoning*. Proc. of Pattern Recognition in Practice IV. Elsevier. 1994.
- Jordan1999 *Learning in Graphical Models*. Ed. By M.I. Jordan. The MIT Press, Cambridge Massachussets. 1999.
- Kalata1984 P.R. Kalata. *The Tracking Index: A Generalized Parameter for α - β and α - β - γ Target Trackers*. IEEE Transactions on Aerospace and Electronic Systems. Vol. 20, No 2. 1984.
- Kalman1960 R.E. Kalman. *A New Approach to Linear Filtering and Prediction Problems*. Journal of Basic Engineering, Transactions ASME, Series D, 82. 1960.
- Kalman1961 R.E. Kalman, R.S. Bucy. *New Results in Linear Filtering and Prediction Theory*. Journal of Basic Engineering, Transactions ASME, Series D, 83. 1961.
- Kawasaki2004 N. Kawasaki, U. Kiencke. *Standard Platform For Sensor Fusion Advanced Driver Assistance System Using Bayesian Networks*. IEEE Intelligent Vehicles Symposium. 2004.
- Kelly1994 A. Kelly. *Essential Kinematics for Autonomous Vehicles*. Technical Report CMU-RI-TR-94-14. Carnegie Mellon University, Pittsburgh. USA. 1994.
- Kim1983 J.H. Kim, J. Pearl. *A Computational Model for Combined Causal and Diagnostic Reasoning in Inference Systems*. Proc. of IJCAI-83. 1983.
- Kim1995 J.H. Kim, M. Valvorta. *On the Detection of Conflicts in Diagnostic Bayesian Networks Using Abstraction*. Proc. of the 11th Annual Conference on UAI. 1995.
- Kitagawa1996 G. Kitagawa. *Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models*. Journal of Computational and Graphical Statistics 5. 1996.

-
- Kjaerulff1993 U. Kjaerulff. *Aspects of Efficiency Improvements in Bayesian Networks*. Thesis. Institute of Electronic Systems. Department of Mathematics and Computer Science. Aalborg University. 1993.
- Kjaerulff1997 U. Kjaerulff. *Nested Junction Trees*. Proc. of the 13th Annual Conference on UAI. 1997.
- Kjaerulff1998 U. Kjaerulff. *Inference in Bayesian Networks Using Nested Junction Trees*. In *Learning in Graphical Models*. Ed. by M.I. Jordan, NATO Science Series, Series D: Behavioural and Social Sciences - Vol. 89. Kluwer Academic Publishers. 1998.
- Klein2004 L.A. Klein. *Sensor and Data Fusion: A tool for Information and Decision Making*. SPIE. 2004.
- Korb2003 K.B. Korb, A.E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC. 2003.
- Kovlov1994 A.V. Kovlov, J.P. Singh. *A Parallel Lauritzen-Spiegelhalter Algorithm for Probabilistic Inference*. Proc. of the 1994 conference on Supercomputing. 1994
- Kunazawa1995 K. Kunazawa, D. Koller, S. Russell. *Stochastic Simulation Algorithms for Dynamic Probabilistic Networks*. Proc. of the 11th International Conference on UAI. 1995.
- Kuo1995 B.C. Kuo. *Automatic Control Systems*. 7th Edition. Prentice-Hall. 1995.
- Lan2002 P. Lan, Q. Ji, C.G. Looney. *Information Fusion with Bayesian Networks for Monitoring Human Fatigue*. Proc. of the 5th International Conference in Information Fusion. 2002.
- Laplante1993 P. Laplante. *Real Time Systems Design and Analysis. An Engineers Handbook*. Ed. IEEE Press. 1993.
- Latombe1991 J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Press. 1991.
- Lauritzen1988 S.L. Lauritzen, D.J. Spiegelhalter. *Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems*. Journal of the Royal Statistical Society, Series B. 1988.
- Lauritzen1992 S.L. Lauritzen. *Propagation of Probabilities, Means and Variances in Mixed Graphical Models*. Journal of the American Statistical Association, 87. 1992.
- Lauritzen1996a S.L. Lauritzen. *Graphical Models*. Oxford Statistical Science Series. Clarendon Press. 1996.

- Lauritzen1996b S.L. Lauritzen, F.V. Jensen. *Local Computation with Valuations for a Communicative Semigroup*. Technical Report R-96-2028. Institute for Electronic Systems. Department of Mathematics and Computer Science. Aalborg University. 1996.
- Lauritzen1999 S.L. Lauritzen, F. Jensen. *Stable Local Computation with Conditional Gaussian Distributions*. Technical Report R-99-2014. Department of Mathematic Sciences, Aalborg University. 1999.
- Lauritzen2002 S.L. Lauritzen. *Some Modern Applications of Graphical Models*. *Highly Structured Stochastic Systems*, Ed. by P.J. Green, N.L. Hjort, S. Richardson. Oxford University Press. 2002.
- Leonard1990 J. Leonard. *Directed Sonar Sensing for Mobile Robot Navigation*. Tesis doctoral. Robotic Research Group. Department of Engineering Science. University of Oxford. 1990.
- Leonard1992 J. Leonard, H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publisher. Boston. 1992.
- Lepar1998 V. Lepar, P. Shenoy. *A Comparison of Lauritzen-Spiegelhalter, Hugin and Shenoy-Shafer Architectures for Computing Marginals of Probability*. Proc. of the 14th International Conference on UAI. 1998.
- Lopez-Orozco1998a J.A. Lopez-Orozco, J.M. de la Cruz, J. Sanz, J. Flores. *Multisensor Fusion Environment Measures Using Bayesian Networks*. Proc. of the International Conference on Multisource-Multisensor Information Fusion. 1998.
- Lopez-Orozco1998b J.A. López-Orozco, J.M. de la Cruz, E. Domínguez, E. Besada, O. R. Polo. *An Open Sensing Architecture to Autonomous Mobile Robots*. Proc. of the IEEE International Symposium on Computational Intelligence in Robotics and Automation. ISCI/CIRA/ISAS Joint Conference. 1998.
- Lopez-Orozco1999 J.A. Lopez-Orozco. *Integración y Fusión Multisensorial de Robots Móviles Autónomos*. Tesis Doctoral. Universidad Complutense Madrid. 1999.
- Lopez-Orozco2000 J.A. Lopez-Orozco, J.M. de la Cruz, E. Besada, P. Rupierez. *An Asynchronous, Robust and Distributed Multisensor Fusion System for Mobile Robots*. The International Journal of Robotics Research. Vol. 19, No 10. 2000.
- Lucas2001 G.W. Lucas. *A tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators*. <http://rosum.sourceforge.net/papers/DiffSteer/DiffSteer.html> 2001.

-
- Luo1989 R.C. Luo, M.G. Kay. *Multisensor Integration and Fusion in Intelligent Systems*. IEEE Trans. on Systems, Man, and Cybernetics. Vol. 19, No 5. 1989.
- Luo1992 R.C. Luo, M.G. Kay. *Data Fusion and Sensor Integration: State-of-art 1990s*. Data Fusion and Machine Intelligence. Ed. by M.A. Abidi, R.C. Gonzalez. Academic Press. 1992.
- Luo1994 R.C. Luo, M.G. Kay. *Multisensor Integration and Fusion for Intelligent Machines and Systems (Computer Engineering and Computer Science)*. Intellect. 2004.
- Lyons1992 D. Lyons. *Planning, Reactive*. Chapter of Encyclopaedia of Artificial Intelligence. S. Shapiro. 1992.
- Madsen1998 A.L. Madsen, F.V. Jensen. *Lazy Propagation in Junction Trees*. Proc. of the 14th Annual Conference on UAI. 1998.
- Madsen1999a A.L. Madsen, F.V. Jensen. *Lazy Propagation: A Junction Tree Inference Algorithm based on Lazy Evaluation*. Artificial Intelligence. 1999.
- Madsen1999b A.L. Madsen, F.V. Jensen. *Parallelization of Inference in Bayesian Networks*. Research Report R-99-5002, Department of Computer Science, Aalborg University, 1999.
- Madsen2000 A.L. Madsen, B. D' Ambrosio. *A Factorized Representation of Independence of Causal Influence and Lazy Propagation*. International Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems. Vol. 8. No 2. 2000.
- Mallick2001 M. Mallick, S. Coraluppi, C. Carthel. *Advances in Asynchronous and Decentralized Estimation*. Proc. IEEE Conf. Aerospace. 2001.
- Mallick2003 M. Mallick, Alan Marrs. *Comparison of the KF and Particle Filter Out-of-Sequence Measurement Filtering Algorithm*. Proc. 6th International Conference on Information Fusion. 2003.
- Mangolini1994 M. Mangolini. *Apport de la fusion d'images satellitaires multicapteurs au niveau pixel en télédétection et photo-interprétation*. Tesis Doctoral, Universidad de Niza-Sophie Antipolis. 1994.
- Manyika1994 J. Manyika, H.F. Durrant-Whyte. *Data Fusion and Sensor Management. A Decentralized Information-Theoretic Approach*. Prentice Hall. 1994.

Bibliografia

- Montemerlo2002 M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem*. Proc. of the AAAI National Conference on Artificial Intelligence. 2002.
- Montemerlo2003 M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit. *FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges*. Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI). 2003.
- Mookerjee2004 P. Mookerjee, F. Reifler. *Application of Reduced State Estimation to Multisensor Fusion with Out of Sequence Measurements*. Proc. IEEE Radar Conference. 2004.
- Müller1996 J. P. Müller. *The Design of Intelligent Agents*. Lecture Notes in Artificial Agents. Springer. 1996.
- MurphyKP1998a K. P. Murphy. *Inference and Learning in Hybrid Bayesian Networks*. Technical Report UCB/CSD-98-990. Computer Science Division, University of California. 1998.
- MurphyKP1998b K.P. Murphy. *Filtering, Smoothing and the Junction Tree Algorithm*. Technical Report, Universidad de California. Berkeley, Department of Computer Science. 1998.
- MurphyKP1998c K.P. Murphy. *Learning switching Kalman Filters*. Compaq Cambridge Research Lab Tech Report 98-10. 1998.
- MurphyKP1999 K.P. Murphy. *Bayesian Map Learning in Dynamic Environments*. Proc. NIPS 1999.
- MurphyRR1998 R.R. Murphy. *Dempster-Shafer Theory for Sensor Fusion in Autonomous Mobile Robots*. IEEE Transactions on Robotics and Automation. Vol. 14. No 2. 1998.
- Mutambara1998 G.O. Mutambara. *Decentralized Estimation and Control for Multisensor Fusion*. CRC Press LLC. 1998.
- Neapolitan1990 R.E. Neapolitan. *Probabilistic Reasoning in Expert Systems. Theory and Algorithms*. John Wiley&Sons Inc. 1990.
- Neapolitan2003 R.E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall. 2003.
- Nettleton2001 E.W. Nettleton and H. Durrant-Whyte. *Delayed and Asequent Data in Decentralized Sensing Networks*. Proc. SPIE Conf #4571. 2001.
- Nicholson1994 A.E. Nicholson, J.M. Brady. *Dynamic Belief Networks for Discrete Monitoring*. IEEE Transactions on System, Man and Cybernetics. Vol. 24, No 11. 1994.

-
- Nilsson1998 D. Nilsson. *An efficient Algorithm for Finding the M Most Probable Configurations in Probabilistic Expert Systems*. Statistics and Computing. 1998.
- Ogata1998 R.C. Ogata. *Modern Control Engineering*. 3rd Edition. Prentice-Hall. 1998.
- Okello2003 N. Okello, G. Thoms. *Thread Assessment Using Bayesian Networks*. Proc. of the 6th International Conference of Information Fusion. 2003.
- Olesen1993 K.G. Olesen. *Causal Probabilistic Networks with Both Discrete and Continuous Variables*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Olesen2002 K.G. Olesen, A.L. Madsen. *Maximal Prime Subgraph Decomposition of Bayesian Networks*. IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 32, No 1. 2002.
- Orton2001 M. Orton, A. Marrs. *A Bayesian Approach to Mutli-target Tracking and Data Fusion with Our-of-Sequence Measurements*. IEE International Workshop on Target Tracking Algorithms and Applications. 2001.
- Pajares2001 G. Pajares, J.M. de la Cruz. *Visión por Computador. Imágenes Digitales y Aplicaciones*. Ed. RA-MA. 2001.
- Pajares2002 G. Pajares, E. Besada-Portas, J.M. de la Cruz. *Analysis of Support Vector Machines and Bayesian Methods for Sunflower Classification*. Recent Research Development in Pattern Recognition. Vol. 3. Transworld Research Network. 2002.
- Pajares2003 G. Pajares, J.M. de la Cruz, E. Besada-Portas, J.A. López-Orozco. *Selección de Propiedades en la Correspondencia Estereoscópica para Sistemas de Visión y Robots Autónomos*. Hispabot 2003.
- Papoulis1990 A. Papoulis. *Probability and Statistics*. Prentice Hall. 1990.
- Pau1988 L.F. Pau. *Sensor Data Fusion*. Journal of Intelligent and Robotics Systems. 1988.
- Payton1991 D. W. Payton. *Internalized Plans: A Representation for Action Resources*. In *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*. Ed. by P. Maes. MIT Press. 1991.
- Pearl1982 J. Pearl. *Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach*. Proc. of American Association for Artificial Intelligence National Conference on AI. 1982.

Bibliografía

- Pearl1986 J. Pearl. *Fusion, Propagation, and Structuring in Belief Networks*. Artificial Intelligence. 1986.
- Pearl1987 J. Pearl. *Evidential Reasoning Using Stochastic Simulation of Causal Models*. Artificial Intelligence. 1987.
- Pearl1988 J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Ed. Morgan Kaufmann. 1988.
- Pearl2000 J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press. 2000.
- Peebles1987 P.Z. Peebles. *Probability, Random Variables, and Random Signal Principles*. McGraw-Hill International Editions. Electrical Engineering Series. 1987.
- Pennock1998 D.M. Pennock. *Logarithmic Time Parallel Bayesian Inference*. Proc. of the 14th Conference on UAI. 1998.
- Pic16F87x *PIC16873/4/6/7 Datasheet*. DS30292C. Microchip Technology Inc. 2001.
- PicMR *PICMicro™ Mid-Range MCU Family Reference Manual*. DS33023A. Microchip Technology Inc. 1997.
- Poole1993 D. Poole. *The Use of Conflicts in Searching Bayesian Networks*. In Preceeding of the 9th Conference on UAI. 1993.
- Regh1999 J.M. Rehg, K.P. Murphy, P.W. Fieguth. *Vision-Based Speaker Detection Using Bayesian Networks*. Computer Vision and Pattern Recognition (CVPR99). 1999.
- Ristic2004 B. Ristic, S. Arulampalam, N. Gordon. *Beyond the Kalman Filter. Particle Filters for Tracking Applications*. Artech House Radar Library. 2004.
- Rodriguez-Polo1998 O. Rodríguez-Polo, J.M. de la Cruz, J.A. Lopez-Orozco, E. Besada-Portas. *Diseño de Practicas de Control Digital Utilizando Multitarea en Tiempo Real y Modelado Orientado a Objetos*. Proc. of XIX Jornadas de Automática CEA-IFAC. 1998.
- Rodriguez-Polo2001a O. Rodríguez-Polo, S. Estaban, A. Maron, L. Grau, J.M. de la Cruz. *Control Code Generator Used for Control Experiments in Ship Scale Models*. In Proc. of CAMS2001 (IFAC Conference). 2001.
- Rodriguez-Polo2001b O. Rodríguez-Polo, J.M. de la Cruz, J.M. Giron-Sierra, S. Esteban. *EdROOM. Automatic C++ Code Generator for Real Systems Modelled UIT ROOM*. Proc. of NTCC2001 (IFAC Conference). 2001.

-
- Rodriguez-Polo2003a O. Rodríguez-Polo. *Generador Automático de Código para Sistemas de Control en Tiempo Real Basado en la Metodología ROOM*. Tesis Doctoral. Universidad Complutense Madrid. 2003.
- Rodriguez-Polo2003b O. Rodríguez Polo, L. del Salvador Carrasco, M. Angulo, J.M. de la Cruz. *Development Plan of the on Board Satellite Software Based on ROOM Modelling and Evolution Component Based Prototypes*. Proc. of the 27th IFAC/IFIP Workshop on Real-Time Programming. 2003.
- Russell1995 S.J. Russell, P. Norving. *Artificial Intelligence. A modern Approach*. Prentice Hall. 1995.
- Russell1999 S. Russell. *Expressive Probability Models for Speech Recognition and Understanding*. Proc. International Workshop on Automatic Speech Recognition and Understanding (ASRU). 1999.
- Sahami1995 M. Sahami, J. Lilly, B. Rollins. *An Autonomous Robot Architecture Using Belief Networks and Neural Networks*. Technical Report University of Stanford. 1995.
- Salmeron2000 A. Salmeron, A. Cano, S. Moral. *Importance Sampling in Bayesian Networks Using Probability Trees*. Computational Statistics and Data Analysis 34. 2000.
- Selic1995 B. Selic, G. Gulleckson, P.T. Ward. *Real-Time Object Oriented Modelling*. John Wiley & Sons. Inc. New York, 1995.
- Shachter1989 R.D. Shachter, C.R. Kenley. *Gaussian Influence Diagrams*. Management Science, 35. 1989.
- Shachter1990a R. Shachter, M. Peot. *Simulation Approaches to General Probabilistic Inference on Bayesian Networks*. In Uncertainty in Artificial Intelligence 5. Ed. by M. Henrion, R.D. Shachter, L.N. Kanal, J.F. Lemmer. 1990.
- Shachter1990b R. Shachter, B. D' Ambrossio, B. Del Favero. *Symbolic Probabilistic Inference in Bayesian Networks*. Proc. of the 8th National Conference on AI. 1990.
- Shafer1976 G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press. 1976.
- Shafer1990a G. Shafer, J. Pearl. *Readings in Uncertainty Reasoning*. Morgan Kauffmann. 1990.
- Shafer1990b G. Shafer, P. Shenoy. *Probability Propagation*. Annals of Mathematics and Artificial Intelligence 2. 1990.

Bibliografia

- Shafer1996 G. Shafer. *Probabilistic Expert Systems*. Society for Industrial and Applied Mathematics. 1996.
- Sherrah2000 J. Sherrah, S. Cong. *Tracking Discontinuous Motion Using Bayesian Inference*. Proc. of the 6th European Conference on Computer Vision. Dublin. 2000.
- Shortliffe1975 E.H. Shortliffe, B.G. Buchanan. *A Model of Inexact Reasoning in Medicine*. Mathematical Biosciences, 23. 1975.
- Simmons1994 R. G. Simmons. *Structured Control for Autonomous Robots*. IEEE Trans. On Robotics and Automation. Vol. 10, No. 1. 1994.
- Singhal1997 A. Singhal, C. Brown. *Dynamic Bayes Net Approach to Multimodal Sensor Fusion*. SPIE Conference on Sensor Fusion and Decentralized Control. 1997.
- Singhal2000 A. Singhal, J. Luo, C. Brown. *A Multilevel Bayesian Network Approach to Image Sensor Fusion*. Proc. of the 3rd International Conference in Information Fusion. 2000.
- Sinopoli2004 B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, S.S. Sastry. *Kalman Filtering with Intermittent Observations*. IEEE Transactions on Automatic Control. Vol. 49. No 9. 2004.
- Sochorova2000 M. Sochorova, J. Vomlel. *Troubleshooting: NP-hardness and Solution methods*. Proc. of the 5th Workshop on Uncertainty Processing (WUPES'2000). 2000.
- Sorenson1970 H.W. Sorenson. *Least-Squares Estimation: from Gauss to Kalman*. IEEE Spectrum. 1970.
- SPIE_Defense&Security *Defense and Security Symposium 2004*. SPIE Proceedings. 2004.
- SPIE_RemoteSensing *Image and Signal Processing for Remote Sensing IX*. SPIE Proceedings. 2004.
- SPIE_SensorFusion *Sensor Fusion. Architectures, Algorithms and Applications VI*. SPIE Proceedings. 2002.
- Srivinikas1996 S. Srivinikas, P. Nayak. *Efficient Enumeration of Instantiations in Bayesian Networks*. Proc. of the 12th Annual Conference on UAI. 1996.
- Theocharous2004 G. Theocharous, K. Murphy, L. Pack Kaelbling. *Representing hierarchical POMDPs as DBNs for multi-scale robot localization*. In Proc. of IEEE Conference on Robotics and Automation (ICRA). 2004.

-
- Thomopoulos1988 S. Thomopoulos, Zhang. *Distributed Filtering with Random Sampling and Delay*. Proc. of the 27th Conference on Decision and Control. 1998.
- Thomopoulos1990 S.C.A. Thomopoulos. *Sensor Integration and Data Fusion*. Journal of Robotic System. Vol. 7. 1990.
- Thrun1997 S. Thrun, W. Burgard, D. Fox. *A Probabilistic Approach for Concurrent Map Acquisition and Localization for Mobile Robots*. Tech-Report School of Computer Science. Carnegie Mellon University. 1997.
- Thrun2002 S. Thrun. *Particle Filters in Robotics*. Proc. of the 18th Annual Conference on UAI. 2002.
- Toledano2003 M. Toledano Luque. *Diseño, construcción y control de un sistema orientable de medida de ultrasonidos capaz de determinar distancias y orientaciones a objetos próximos*. Proyecto fin de carrera de Ingeniería Electrónica, Departamento de Arquitectura y Automática, UCM. 2003.
- Torralba2003 A. Torralba, K.P. Murphy, W.T. Freeman, M.A. Rubin. *Context-Based Vision System for Place and Object Recognition*. AI Memo 2003-05. MIT. 2003.
- Trebi-Ollenum1999 A. Trebi-Ollenu, J.M. Dollan. *An Autonomous Ground Vehicle for Distributed Surveillance: CyberScout*. Internal Technical Report, ICES, Carnegie Mellon University. 1999.
- UAI2004 *Proc. of the 20th Conference on Uncertainty in Artificial Intelligence*. 2004. (www.auai.org)
- Vapnik1998 V.N. Vapnik. *Statistical Learning Theory*. Wiley. 1998.
- Vapnik2000 V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag. 2000.
- Varshney1997 P.K. Varshney. *Distributed Detection and Data Fusion*. Springer-Verlag. 1997.
- Vazquez2000 C. Vázquez Vélez. *Módulo de calibración de cámaras para visión artificial*. Proyecto fin de carrera de Ingeniería Electrónica, Departamento de Arquitectura y Automática, UCM. 2000.
- Wald1998 L. Wald. A European Proposal for terms of reference in data fusion. *International Archives of Photogrammetry and Remote Sensing*. Vol. 32. 1998.
- Wald1999 L. Wald. *Some terms of reference in data fusion*. IEEE Transactions on Geosciences and Remote Sensing. 1999.

Bibliografia

- Waltz1990 E.L. Waltz, J. Llinas. *Multisensor Data Fusion*. Artech House, Boston. 1990.
- White1988 F.E. White. *A model for Data Fusion*. Proc. of the 1st National Symposium on Sensor Fusion. 1988.
- White1991 F. White. *Data Fusion Lexicon*. Data Fusion Subpanel for C3. Code 4202. 1991.
- Willsky2002 A.S. Willsky. *Multiresolution Models for Signal and Image Processing*. Proc. of the IEEE, Vol. 90. No 8. 2002.
- Xiang2000 Y. Xiang, C.G. Olesen, F.V. Jensen. *Practical Issues in Modeling Large Diagnostic Systems with Multiply Sectioned Bayesian Networks*. International Journal of Pattern Recognition and Artificial Intelligence. 2000.
- Zadeh1975 L.A. Zadeh. *Fuzzy Logic and Approximate Reasoning*. Synthese. No 30. 1975.
- Zadeh1978 L.A. Zadeh. *Fuzzy Sets as a Basis for a Theory of Possibility*. Fuzzy Sets and Systems, Vol. 1, No 1. 1978.
- ZhangK2002 K. Zhang, X. Li, Y. Zhu. *Optimal Update with Out-of-Sequence Measurements for Distributed Filtering*. Proc. 5th International Conference on Information Fusion. 2002.
- ZhangK2003 K. Zhang, X. Li, H. Chen. *Multi-Sensor Multi-Target Tracking with Out of Sequence Measurements*. Proc. 6th International Conference on Information Fusion. 2003.
- ZhangNL1996 N.L. Zhang. *Exploiting Causal Independence in Bayesian Networks Inference*. Journal of Artificial Intelligence Research. Vol. 5. 1996.
- ZhangY2002 Y. Zhang, Q. Ji, C.G. Looney. *Active Information Fusion for Decision Making Under Uncertainty*. Proc. of the 5th International Conference in Information Fusion. 2002.
- Zhou2003 H. Zhou, S. Sakane. *Learning Bayesian Network Structure for Environment and Sensor Planning for Mobile Robot Localization*. IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems. 2003.
- Zhu2002 Y. Zhu. *Multisensor Decision and Estimation Fusion*. Kluwer Academic Publishers. 2002.

Apéndice I

Expresiones del filtro de Kalman y del filtro de Información

En este apéndice se recopilan las expresiones del filtro de Kalman y del filtro de Información para estimar el estado y covarianza de cuatro tipos de sistemas de control diferentes: sistemas lineales con un único sensor, sistemas lineales multisensoriales, sistemas no lineales con un único sensor, y sistemas no lineales multisensoriales. Para cada tipo de sistema, se presentan, dentro de una misma tabla y de forma enfrentada, las expresiones de cada uno de los filtros, agrupadas de acuerdo con las tres etapas habituales (predicción, validación y asimilación) que se realizan en cada ciclo de filtrado. Con esta organización se facilita una comparación directa de las expresiones correspondientes a cada filtro para el mismo tipo de sistema, y un análisis indirecto, por medio del uso simultáneo de varias tablas, de las expresiones asociadas a cada etapa y filtro para cada tipo de sistema. Además, con el objetivo de poder comparar fácilmente las características de los cuatro tipos de sistemas de control, se incluye una tabla adicional que presenta de forma conjunta, para todos los sistemas, las expresiones que modelan el comportamiento del sistema y de su (sus) sensor (sensores).

Para cada uno de los tipos de sistema, los resultados obtenidos por los dos filtros son algebraicamente equivalentes, aunque no iguales, ya que el espacio en el que operan es diferente, y por lo tanto es necesario realizar una conversión de un espacio de trabajo al otro antes de poder realizar cualquier tipo de comparación. Un análisis de la equivalencia algebraica de las expresiones para las etapas de predicción y asimilación puede encontrarse en [Mutambara1998]. La demostración para las tres etapas es fácil de hacer, teniendo siempre en cuenta que el filtro de Kalman opera en el espacio de estados del sistema, mientras que el filtro de Información lo hace en el espacio de la información existente sobre el estado del sistema; y que los dos espacios de trabajo se encuentran relacionados por la matriz de covarianza sobre el estado del sistema, que en su forma inversa es la matriz de covarianza sobre la información del estado del sistema. Los espacios de trabajo de cada filtro y las expresiones que lo relacionan se presentan en una tabla adicional.

La diferente complejidad operacional existente para cada etapa entre el filtro de Kalman y en el filtro de Información es una consecuencia directa de los distintos espacios de trabajo en los que opera cada filtro. En la etapa de predicción se estima el valor estado a partir del valor estimado del estado para el instante de muestreo anterior, por lo que la complejidad de esta operación es menor cuando es realizada en el espacio de estados, y por lo tanto, dicha etapa es más sencilla en el filtro de Kalman. La etapa de asimilaciones es utilizada para añadir al valor estimado del estado, la nueva información sensorial, por lo que la complejidad de esta operación es menor cuando se realiza en el espacio de información sobre el estado, y por lo tanto, dicha etapa es más sencilla en el filtro de Información. La etapa opcional de validación, en la que se decide si la medida debe de ser asimilada o desechada, se realiza en un nuevo espacio de trabajo: el espacio de medida. Aunque dicho espacio de trabajo se encuentra relacionado simultáneamente con el espacio de estado y el espacio de información sobre el estado, la relación más sencilla se establece, a partir de las ecuaciones que implementan el modelo de medida, con el espacio de estado. Además, hay que

tener en cuenta que en los sistemas de control no lineales, y como consecuencia directa de la falta de linealidad en las funciones que modelan el comportamiento del sistema y el de los sensores, todas las etapas de ambos filtros se encuentran directamente relacionadas con el espacio de estado del sistema, por lo que en el caso del filtro de Información en todas las etapas será necesario convertir el valor estimado sobre la información del estado del sistema en el valor estimado sobre el estado del sistema.

Por último, es importante reseñar que a pesar de la equivalencia algebraica de ambos filtros, a la hora de implementarlos y ejecutarlos, éstos no obtienen resultados idénticos, debido a que las diferencias operacionales existentes en la implementación de cada filtro introducen errores numéricos computacionales diferentes. Además de los errores de filtrado introducidos en cada caso, estarán los atribuibles al simple hecho de la conversión final entre los dos espacios de trabajo, operación necesaria para poder realizar la comparación de los resultados. Estas diferencias se pueden hacer despreciables eligiendo correctamente las unidades de las magnitudes de las variables sobre las que se trabaja. De no ser así, cabría esperar que ninguno de los dos filtros esté trabajando correctamente, ya que en alguna de sus etapas se estarían introduciendo errores numéricos computacionales no despreciables.

El contenido de este apéndice se encuentra dividido en dos secciones. En la primera se presentan los cuatro tipos de sistemas de control, y se analizan las diferencias existentes entre los mismos. En la segunda se exponen las características principales del filtro de Kalman y del filtro de Información, las relaciones existentes entre ellos, y las ecuaciones que implementan las etapas fundamentales de cada filtro para cada uno de los cuatro sistemas.

A1.1. Los cuatro tipos de sistemas de control propuestos

Las expresiones del filtro de Kalman y del filtro de Información recopiladas en este apéndice son utilizadas para estimar el estado y covarianza de cuatro tipos de sistemas de control discretos diferentes: sistemas lineales con un único sensor, sistemas lineales multisensoriales, sistemas no lineales con un único sensor, y sistemas no lineales multisensoriales.

Para los cuatro casos, el estado del sistema \mathbf{x}_{t+1} en el instante $t+1$ depende del estado \mathbf{x}_t en el instante de muestreo anterior t y de la señal de control \mathbf{u}_t aplicada durante el periodo de muestreo. En el caso de un único sensor, el estado \mathbf{x}_{t+1} en el instante $t+1$ es observado por un sensor que proporciona en ese instante la medida \mathbf{z}_{t+1} , mientras que en el caso multisensorial, el estado \mathbf{x}_{t+1} en el instante $t+1$ es observado por un conjunto de S sensores que proporcionan en ese instante las medidas $\mathbf{z}_{s,t+1}$. Según el tipo de sistema, las relaciones existentes entre el estado del sistema \mathbf{x}_{t+1} en el instante $t+1$ y el estado del sistema \mathbf{x}_t en el instante anterior t y la señal de control \mathbf{u}_t aplicada durante el periodo de muestreo, y entre el estado del sistema \mathbf{x}_{t+1} y la (las) medida (medidas) \mathbf{z}_{t+1} ($\mathbf{z}_{s,t+1}$) realizada (realizadas) dentro del mismo periodo de muestreo $t+1$, pueden ser modeladas por medio de funciones lineales o no lineales. Para todos los casos, se modela el desconocimiento exacto del estado del sistema y de la medida, por medio de un ruido blanco aditivo con media $\mathbf{0}$ y covarianza \mathbf{Q}_t para el modelo del sistema y \mathbf{R}_{t+1} ($\mathbf{R}_{s,t+1}$) para el (los) modelo (modelos) sensorial (sensoriales). Las expresiones que modelan los cuatro tipos de sistema de control se presentan en la tabla A1-1.

	Un Sensor	Múltiples Sensores
Sistemas Lineales	<p>MS: $\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{u}_t + \mathbf{v}_t \quad N_{\mathbf{v}_t}(\mathbf{0}, \mathbf{Q}_t)$</p> <p>MM: $\mathbf{z}_{t+1} = \mathbf{H}_{t+1} \mathbf{x}_{t+1} + \mathbf{v}_{t+1} \quad N_{\mathbf{v}_{t+1}}(\mathbf{0}, \mathbf{R}_{t+1})$</p> <p>Se desea estimar $\hat{\mathbf{x}}_{n\data_n}$ y $\mathbf{P}_{n\data_n}$ con</p> <p>$data_n = \left\{ \hat{\mathbf{x}}_0, \mathbf{P}_0; \mathbf{z}_t = \zeta_t, \mathbf{H}_t, \mathbf{R}_t, \mathbf{u}_{t-1}, \mathbf{F}_{t-1}, \right\}$</p> <p>$\mathbf{Q}_{t-1}, \text{ con } t = 1 : n$</p>	<p>MS: $\mathbf{x}_{t+1} = \mathbf{F}_t \mathbf{x}_t + \mathbf{u}_t + \mathbf{v}_t \quad N_{\mathbf{v}_t}(\mathbf{0}, \mathbf{Q}_t)$</p> <p>MM: $\mathbf{z}_{s,t+1} = \mathbf{H}_{s,t+1} \mathbf{x}_{t+1} + \mathbf{v}_{s,t+1} \quad N_{\mathbf{v}_{s,t+1}}(\mathbf{0}, \mathbf{R}_{s,t+1})$</p> <p>Se desea estimar $\hat{\mathbf{x}}_{n\data_n}$ y $\mathbf{P}_{n\data_n}$ con</p> <p>$data_n = \left\{ \hat{\mathbf{x}}_0, \mathbf{P}_0; \mathbf{z}_{s,t} = \zeta_{s,t}, \mathbf{H}_{s,t}, \mathbf{R}_{s,t}, \mathbf{u}_{t-1}, \mathbf{F}_{t-1}, \right\}$</p> <p>$\mathbf{Q}_{t-1}, \text{ con } t = 1 : n, s = 1 : S$</p>
	<p>MS: $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, t) + \mathbf{v}_t \quad N_{\mathbf{v}_t}(\mathbf{0}, \mathbf{Q}_t)$</p> <p>MM: $\mathbf{z}_{t+1} = h(\mathbf{x}_{t+1}, t) + \mathbf{v}_{t+1} \quad N_{\mathbf{v}_{t+1}}(\mathbf{0}, \mathbf{R}_{t+1})$</p> <p>Se desea estimar $\hat{\mathbf{x}}_{n\data_n}$ y $\mathbf{P}_{n\data_n}$ con</p> <p>$data_n = \left\{ \hat{\mathbf{x}}_0, \mathbf{P}_0; \mathbf{z}_t = \zeta_t, h(), \mathbf{R}_t, \mathbf{u}_{t-1}, f(), \right\}$</p> <p>$\mathbf{Q}_{t-1}, \text{ con } t = 1 : n$</p>	<p>MS: $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, t) + \mathbf{v}_t \quad N_{\mathbf{v}_t}(\mathbf{0}, \mathbf{Q}_t)$</p> <p>MM: $\mathbf{z}_{s,t+1} = h_s(\mathbf{x}_{t+1}, t) + \mathbf{v}_{s,t+1} \quad N_{\mathbf{v}_{s,t+1}}(\mathbf{0}, \mathbf{R}_{s,t+1})$</p> <p>Se desea estimar $\hat{\mathbf{x}}_{n\data_n}$ y $\mathbf{P}_{n\data_n}$ con</p> <p>$data_n = \left\{ \hat{\mathbf{x}}_0, \mathbf{P}_0; \mathbf{z}_{s,t} = \zeta_{s,t}, h_s(), \mathbf{R}_{s,t}, \mathbf{u}_{t-1}, f(), \right\}$</p> <p>$\mathbf{Q}_{t-1}; \text{ con } t = 1 : n, s = 1 : S$</p>

Tabla A1-1. Ecuaciones que modelan el comportamiento del sistema (MS) y el comportamiento de los sensores (MM) para los cuatro tipos de sistemas de control que se analizan (sistemas lineales con un sensor, sistemas lineales multisensoriales, sistemas no lineales (NL) con un sensor, sistemas no lineales (NL) multisensoriales). En los cuatro casos se desea estimar el estado del sistema $\hat{\mathbf{x}}_{n\data_n}$ y su covarianza

$\mathbf{P}_{n\data_n}$ en el instante n a partir de toda la información en $data_n$.

El modelo del sistema también se denomina función/ecuación de transición de estados, mientras que el modelo de medida también es conocido como función/ecuación de medida. La matriz \mathbf{F}_t que aparece en la ecuación de transición lineal se denomina matriz de transición de estados ya que es la responsable de modelar la relación existente entre el estado del sistema entre dos momentos de muestreo consecutivos. De forma análoga, la matriz \mathbf{H}_{t+1} ($\mathbf{H}_{s,t+1}$) que aparece en la ecuación de medida lineal se denomina matriz de medida.

Es importante destacar que los sistemas multisensoriales pueden ser modelados como sistemas monosensoriales, concatenando en un único vector de medidas los vectores de medidas procedentes de los distintos sensores ($\mathbf{z}_{t+1}=[\mathbf{z}_{1,t+1};\mathbf{z}_{2,t+1};\dots;\mathbf{z}_{S,t+1}]$), en una única función modelo las funciones modelo de cada sensor ($\mathbf{H}_{t+1}=[\mathbf{H}_{1,t+1};\mathbf{H}_{2,t+1};\dots;\mathbf{H}_{S,t+1}]$ para el caso lineal y $h(x,t)=[h_1;h_2;\dots;h_S](x,t)$ para el caso no lineal), y en una única covarianza las covarianzas asociadas a cada sensor ($\mathbf{R}_{t+1}=[\mathbf{R}_{1,t+1},\mathbf{0},\dots,\mathbf{0};\mathbf{0},\mathbf{R}_{2,t+1},\dots,\mathbf{0};\dots;\mathbf{0},\dots,\mathbf{R}_{S,t+1}]$).

Sin embargo, la operación inversa, tratar un sistema monosensorial como un sistema multisensorial, no siempre es posible, ya que para ello es necesario poder descomponer el espacio de medida del sensor en subconjuntos probabilísticamente independientes entre sí conocido el estado del sistema, o lo que es lo mismo, que los ruidos asociados a los subconjuntos de medida no se encuentren correlados.

A1.2. El filtro de Kalman y del filtro de Información

El filtro de Kalman (KF) es un estimador lineal recursivo que, en función del modelo de la dinámica de un sistema, su modelo sensorial, y de un conjunto de observaciones (medidas) periódicas, obtiene, cíclicamente, el valor estimado del estado de un sistema $\hat{\mathbf{x}}_{n\data_n}$ y su covarianza $\mathbf{P}_{n\data_n}$, de forma que se minimice el error cuadrático medio entre el valor real y el valor estimado. En su formulación original, se supone que tanto el modelo del sistema como el modelo de medida son funciones lineales, y que el ruido en el modelo del sistema y en el modelo de medida es una variable gaussiana con media cero y una determinada covarianza. Cuando las relaciones no se pueden modelar con funciones lineales, se puede optar bien por linealizar las funciones no lineales, o por utilizar la versión extendida del filtro de Kalman para el caso no lineal, el filtro de Kalman Extendido (EKF). En ambos casos, los dos filtros (KF y EKF) operan en el espacio del estado del sistema.

El filtro de Información (IF) es esencialmente el filtro de Kalman expresado en términos de la información que se tiene sobre el estado del sistema $\hat{\mathbf{y}}_{n\data_n}$ y la covarianza $\mathbf{Y}_{n\data_n}$ asociada a dicha información. De la misma forma, el filtro Extendido de Información (EIF) es la versión asociada al filtro Extendido de Kalman (EKF) que trabaja en el espacio de la información existente sobre el estado del sistema.

La relación existente entre los dos espacios de trabajo, viene fijada por la matriz de covarianza del estado del sistema, que en su forma inversa es la matriz de covarianza sobre la información disponible sobre el estado del sistema. De esta forma, y como cabría esperar, cuando aumenta el valor de la matriz de covarianza del estado, y por lo tanto el desconocimiento sobre el estado real del sistema, decrece la información existente sobre el estado del sistema, y viceversa. Para realizar el cambio de espacio de trabajo, únicamente hay que multiplicar la variable de ese espacio por la inversa de su covarianza. En la tabla A1-2 se presentan los espacios de trabajo para los cuatro filtros y las expresiones necesarias para la conversión de los espacios equivalentes (KF equivalente a IF, y EKF equivalente a EIF).

	KF y EKF	IF y EIF
Espacio de Trabajo	$\hat{\mathbf{x}}_{n\data_n}$ y $\mathbf{P}_{n\data_n}$	$\hat{\mathbf{y}}_{n\data_n}$ e $\mathbf{Y}_{n\data_n}$
Relaciones	$\hat{\mathbf{x}}_{k/j} = \mathbf{Y}_{k/j}^{-1} \hat{\mathbf{y}}_{k/j}$ $\mathbf{P}_{k/j} = \mathbf{Y}_{k/j}^{-1}$	$\hat{\mathbf{y}}_{k/j} = \mathbf{P}_{k/j}^{-1} \hat{\mathbf{x}}_{k/j}$ $\mathbf{Y}_{k/j} = \mathbf{P}_{k/j}^{-1}$

Tabla A1-2. Espacios de trabajo del KF, del EKF, del IF, y del EIF, junto con las expresiones que relacionan los espacios equivalentes (KF equivalente a IF, y EKF equivalente a EIF). El valor de $data_n$ para cada uno de los cuatro tipos de sistema puede encontrarse en la tabla A1-1.

Como consecuencia directa de los espacios de trabajo en los que operan los diferentes filtros, la complejidad de las operaciones realizadas en cada etapa para cada filtro es diferente. En las secciones siguientes se comenta, para cada etapa de filtrado, las diferencias computacionales existentes entre los filtros asociados a cada uno de los cuatro tipos de sistemas. Por último se presentan las ecuaciones de las tres etapas de filtrado para los filtros de Kalman (KF o EKF) y de Información (IF o EIF) para los cuatro tipos de sistemas en las tablas A1-7, A1-8, A1-9 y A1-10. En cada caso, la etapa de menor complejidad es destacada sombreando la celda en la que se encuentra.

A1.2.1. Predicción

Durante la etapa de predicción se utiliza el modelo del sistema para estimar el valor del estado del sistema $\hat{\mathbf{x}}_{t+1/t}$ a partir del valor estimado en el instante anterior $\hat{\mathbf{x}}_{t/t}$ y la señal de control \mathbf{u}_t . Las expresiones utilizadas para la predicción en los cuatro tipos de sistemas se presentan en la tabla A1-2.

La complejidad de esta operación es mínima cuando se realiza en el espacio del estado del sistema, es decir para las dos versiones de filtros de Kalman (KF y EKF). En los filtros de Información (IF y EIF) es necesario realizar una doble conversión de espacios: primero se realiza una conversión del espacio de información al de estado, luego se realiza la predicción y por último se retorna al espacio de información.

Sistema (versión del filtro)	Filtro Kalman	Filtro de Información
Lineal (Filtro Lineal)	$\hat{\mathbf{x}}_{t+1/t} = \mathbf{F}_t \hat{\mathbf{x}}_{t/t} + \mathbf{u}_t$ $\mathbf{P}_{t+1/t} = \mathbf{F}_t \mathbf{P}_{t/t} \mathbf{F}_t^T + \mathbf{Q}_t$	$\hat{\mathbf{y}}_{t+1/t} = \mathbf{Y}_{t+1/t} (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \hat{\mathbf{y}}_{t/t} + \mathbf{u}_t)$ $\mathbf{Y}_{t+1/t} = (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \mathbf{F}_t^T + \mathbf{Q}_t)^{-1}$
No lineal (Filtro Extendido)	$\mathbf{F}_t = (\mathbf{J}f)_{\mathbf{x}}(\hat{\mathbf{x}}_{t/t}, \mathbf{u}_t, t)$ $\hat{\mathbf{x}}_{t+1/t} = f(\hat{\mathbf{x}}_{t/t}, \mathbf{u}_t, t)$ $\mathbf{P}_{t+1/t} = \mathbf{F}_t \mathbf{P}_{t/t} \mathbf{F}_t^T + \mathbf{Q}_t$	$\hat{\mathbf{x}}_{t/t} = \mathbf{Y}_{t/t}^{-1} \hat{\mathbf{y}}_{t/t}$ $\mathbf{F}_t = (\mathbf{J}f)_{\mathbf{x}}(\hat{\mathbf{x}}_{t/t}, \mathbf{u}_t, t)$ $\hat{\mathbf{y}}_{t+1/t} = \mathbf{Y}_{t+1/t} f(\hat{\mathbf{x}}_{t/t}, \mathbf{u}_t, t)$ $\mathbf{Y}_{t+1/t} = (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \mathbf{F}_t^T + \mathbf{Q}_t)^{-1}$

Tabla A1-3. Ecuaciones para la etapa de predicción del KF, IF, EKF y EIF

Además de actualizar el valor estimado del estado (o de la información sobre el estado), durante esta etapa se actualiza el valor de la covarianza del estado (o de su información). En el caso de los filtros extendidos, es decir cuando el modelo del sistema es no lineal, la matriz \mathbf{F}_t utilizada durante la actualización de la covarianza se obtiene como la derivada primera (jacobiano) respecto a la variable de estado \mathbf{x} de la función no lineal $f(\mathbf{x}, \mathbf{u}, t)$, evaluada para el instante t sobre el valor estimado del estado $\hat{\mathbf{x}}_{t/t}$ y la señal de control \mathbf{u}_t . Este hecho no supone una sobrecarga adicional para el filtro de Información, ya que la conversión del espacio de la información al de estado es necesaria en todos los casos.

A1.2.2. Asimilación

Durante la etapa de asimilación se utiliza la (las) medida (medidas) ζ_{t+1} ($\zeta_{s,t+1}$) proporcionada (proporcionadas) por el (los) sensor (sensores) para actualizar el valor estimado del estado durante la etapa de predicción. Es decir, en esta etapa se añade nueva información sobre el estado del sistema a la información que ya se tenía.

La complejidad de las operaciones es mínima para los filtros que operan en el espacio de la información que se tiene sobre el estado, es decir para las dos versiones de filtro de Información (IF y EIF). En los filtros de Kalman (KF y EKF) se añade al valor estimado del estado un término, de cálculo más complejo, que mide la discrepancia entre la medida realizada y el valor estimado de la medida (a partir del valor estimado sobre el estado en la etapa de predicción). El aumento de la complejidad de la etapa para Kalman se hace todavía más patente en los sistemas multisensoriales.

Las diferencias en las operaciones realizadas en esta etapa para los distintos tipos de sistema de control propuestos son significativas, por lo que se comentan a continuación.

▪ Sistemas lineales con un único sensor

Las ecuaciones de asimilación del KF y del IF para este tipo de sistemas se recogen en la siguiente tabla. En la misma se puede observar que los cálculos que hay que realizar en el filtro de

Información tienen una complejidad menor, ya que tras obtener el término común $\mathbf{H}_{t+1}^T \mathbf{R}_{t+1}^{-1}$ que aparece en la información de medida \mathbf{i}_{t+1} , y en su covarianza \mathbf{I}_{t+1} , sólo es necesario realizar dos productos de matrices y dos sumas. Además, en los modelos de medida invariantes en el tiempo (con \mathbf{H}_{t+1} y \mathbf{R}_{t+1} constantes) los cálculos se simplifican más, ya que se puede calcular al inicio del algoritmo $\mathbf{H}_{t+1}^T \mathbf{R}_{t+1}^{-1}$ y la covarianza de la información sensorial \mathbf{I}_{t+1} .

Filtro de Kalman	Filtro de Información
$\hat{\mathbf{r}}_{t+1/t} = \zeta_{t+1} - \mathbf{H}_{t+1} \hat{\mathbf{x}}_{t+1/t}$ $\mathbf{S}_{t+1/t} = \mathbf{H}_{t+1} \mathbf{P}_{t+1/t} \mathbf{H}_{t+1}^T + \mathbf{R}_{t+1}$ $\mathbf{W}_{t+1/t} = \mathbf{P}_{t+1/t} \mathbf{H}_{t+1}^T \mathbf{S}_{t+1/t}^{-1}$ $\hat{\mathbf{x}}_{t+1/t+1} = \hat{\mathbf{x}}_{t+1/t} + \mathbf{W}_{t+1/t} \hat{\mathbf{r}}_{t+1/t}$ $\mathbf{P}_{t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1/t} \mathbf{S}_{t+1/t} \mathbf{W}_{t+1/t}^T$	$\mathbf{i}_{t+1} = \mathbf{H}_{t+1}^T \mathbf{R}_{t+1}^{-1} \zeta_{t+1}$ $\mathbf{I}_{t+1} = \mathbf{H}_{t+1}^T \mathbf{R}_{t+1}^{-1} \mathbf{H}_{t+1}$ $\hat{\mathbf{y}}_{t+1/t+1} = \hat{\mathbf{y}}_{t+1/t} + \mathbf{i}_{t+1}$ $\mathbf{Y}_{t+1/t+1} = \mathbf{Y}_{t+1/t} + \mathbf{I}_{t+1}$

Tabla A1-4. Etapa de asimilación para sistemas lineales con un único sensor.

En el caso del filtro de Kalman, la complejidad es siempre mayor, y aún en el caso en el que el modelo de medida sea invariante, en todas las etapas es necesario calcular las matrices $\mathbf{S}_{t+1/t}$ y $\mathbf{W}_{t+1/t}$ antes de poder actualizar el estado del sistema y su covarianza, ya que estas variables dependen de la covarianza del estado del sistema.

▪ **Sistemas lineales multisensoriales**

Las ecuaciones de asimilación del KF y del IF para este tipo de sistemas se recogen en la siguiente tabla. En la misma se puede observar que la complejidad para el filtro de Información es mucho menor, porque en este espacio es posible realizar directamente la partición de toda la información sensorial (\mathbf{i}_{t+1} e \mathbf{I}_{t+1}) en subconjuntos que almacenan la información sensorial asociada a cada sensor ($\mathbf{i}_{s,t+1}$ e $\mathbf{I}_{s,t+1}$), al ser $\mathbf{R}_{t+1} = [\mathbf{R}_{1,t+1}, \mathbf{0}, \dots, \mathbf{0}; \mathbf{0}, \mathbf{R}_{2,t+1}, \dots, \mathbf{0}; \dots; \mathbf{0}, \dots, \mathbf{R}_{S,t+1}]$, $\mathbf{H}_{t+1} = [\mathbf{H}_{1,t+1}; \mathbf{H}_{2,t+1}; \dots; \mathbf{H}_{S,t+1}]$, y $\zeta_{t+1} = [\zeta_{1,t+1}; \zeta_{2,t+1}; \dots; \zeta_{S,t+1}]$.

Filtro de Kalman	Filtro de Información
$\hat{\mathbf{r}}_{s,t+1/t} = \zeta_{s,t+1} - \mathbf{H}_{s,t+1} \hat{\mathbf{x}}_{t+1/t}$ $\mathbf{S}_{s,t+1/t} = \mathbf{H}_{s,t+1} \mathbf{P}_{t+1/t} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ $\mathbf{W}_{s,t+1/t} = \mathbf{P}_{t+1/t} \mathbf{H}_{s,t+1}^T \mathbf{S}_{s,t+1/t}^{-1}$ $\tilde{\mathbf{x}}_{s,t+1/t+1} = \hat{\mathbf{x}}_{t+1/t} + \mathbf{W}_{s,t+1/t} \hat{\mathbf{r}}_{s,t+1/t}$ $\tilde{\mathbf{P}}_{s,t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{s,t+1/t} \mathbf{S}_{s,t+1/t} \mathbf{W}_{s,t+1/t}^T$ $\mathbf{c}_x = \sum_{s=1}^S (\tilde{\mathbf{P}}_{s,t+1/t+1}^{-1} \tilde{\mathbf{x}}_{s,t+1/t+1} - \mathbf{P}_{t+1/t}^{-1} \hat{\mathbf{x}}_{t+1/t})$ $\hat{\mathbf{x}}_{t+1/t+1} = \mathbf{P}_{t+1/t+1} (\mathbf{P}_{t+1/t}^{-1} \hat{\mathbf{x}}_{t+1/t} + \mathbf{c}_x)$ $\mathbf{P}_{t+1/t+1} = (\mathbf{P}_{t+1/t}^{-1} + \sum_{s=1}^S (\tilde{\mathbf{P}}_{s,t+1/t+1}^{-1} - \mathbf{P}_{t+1/t}^{-1}))^{-1}$	$\mathbf{i}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}$ $\mathbf{I}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1}$ $\hat{\mathbf{y}}_{t+1/t+1} = \hat{\mathbf{y}}_{t+1/t} + \sum_{s=1}^S \mathbf{i}_{s,t+1}$ $\mathbf{Y}_{t+1/t+1} = \mathbf{Y}_{t+1/t} + \sum_{s=1}^S \mathbf{I}_{s,t+1}$

Tabla A1-5. Etapa de asimilación para sistemas lineales multisensoriales.

En el filtro de Kalman no es posible realizar una descomposición de este tipo, y por lo tanto, la complejidad de esta etapa aumenta considerablemente, pudiéndose distinguir dos partes dentro de la misma. En la primera, similar a la del caso monosensorial pero realizada para cada uno de los sensores, se obtienen las matrices $\mathbf{S}_{s,t+1/t}$ y $\mathbf{W}_{s,t+1/t}$, y una nueva estima del estado del sistema $\hat{\mathbf{x}}_{s,t+1/t+1}$ en función de la discrepancia entre la medida $\zeta_{s,t+1}$ y el valor estimado de medida $\mathbf{H}_{s,t+1}\hat{\mathbf{x}}_{s,t+1/t}$. En la segunda se actualiza el valor del estado del sistema a partir de las discrepancias \mathbf{c}_x existentes, en el espacio de información, entre el valor estimado del estado del sistema tras la predicción y los valores estimados del estado del sistema para cada sensor.

▪ **Sistemas no lineales con un único sensor**

Los casos monosensorial lineal y no lineal se diferencian en las funciones del modelo de medida: $\mathbf{H}_{t+1}\mathbf{x}_{t+1}$ en el primero y $h(x,t)$ en el segundo. Las ecuaciones de asimilación del EKF y del EIF para este tipo de sistemas se recogen en la siguiente tabla.

Filtro de Kalman Extendido	Filtro de Información Extendido
$\mathbf{H}_{t+1} = (\mathbf{J}h)_x(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\hat{\mathbf{r}}_{t+1/t} = \zeta_{t+1} - h(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\mathbf{S}_{t+1/t} = \mathbf{H}_{t+1}\mathbf{P}_{t+1/t}\mathbf{H}_{t+1}^T + \mathbf{R}_{t+1}$ $\mathbf{W}_{t+1/t} = \mathbf{P}_{t+1/t}\mathbf{H}_{t+1}^T\mathbf{S}_{t+1/t}^{-1}$ $\hat{\mathbf{x}}_{t+1/t+1} = \hat{\mathbf{x}}_{t+1/t} + \mathbf{W}_{t+1/t}\hat{\mathbf{r}}_{t+1/t}$ $\mathbf{P}_{t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1/t}\mathbf{S}_{t+1/t}\mathbf{W}_{t+1/t}^T$	$\hat{\mathbf{x}}_{t+1/t} = \mathbf{Y}_{t+1/t}^{-1}\hat{\mathbf{y}}_{t+1/t}$ $\mathbf{H}_{t+1} = (\mathbf{J}h)_x(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\zeta_{t+1}^C = \zeta_{t+1} - h(\hat{\mathbf{x}}_{t+1/t}, t+1) + \mathbf{H}_{t+1}\hat{\mathbf{x}}_{t+1/t}$ $\mathbf{i}_{t+1} = \mathbf{H}_{t+1}^T\mathbf{R}_{t+1}^{-1}\zeta_{t+1}^C$ $\mathbf{I}_{t+1} = \mathbf{H}_{t+1}^T\mathbf{R}_{t+1}^{-1}\mathbf{H}_{t+1}$ $\hat{\mathbf{y}}_{t+1/t+1} = \hat{\mathbf{y}}_{t+1/t} + \mathbf{i}_{t+1}$ $\mathbf{Y}_{t+1/t+1} = \mathbf{Y}_{t+1/t} + \mathbf{I}_{t+1}$

Tabla A1-6. Etapa de asimilación para sistemas no lineales con un único sensor

El EKF tiene en cuenta dicha diferencia, sustituyendo para la etapa de asimilación los modelos de medida e igualando las \mathbf{H}_{t+1} por la derivada primera (jacobiano) respecto a la variable de estado \mathbf{x} de la $h(x,t)$, evaluada para el instante $t+1$ sobre el valor estimado del estado $\hat{\mathbf{x}}_{t+1/t}$.

En el EIF, además de los cambios expuestos para el caso del EKF, en el cálculo de la información de medida $\mathbf{i}_{s,t+1}$ se utiliza una medida corregida $\zeta_{t+1}^C = (\zeta_{t+1} - h(\mathbf{Y}_{t+1/t}^{-1}\hat{\mathbf{y}}_{t+1/t}, t+1) + \mathbf{H}_{t+1}\mathbf{Y}_{t+1/t}^{-1}\hat{\mathbf{y}}_{t+1/t})$ en vez de la medida real ζ_{t+1} . Esta medida corregida recoge, además de la información proporcionada por el sensor, la discrepancia existente entre la predicción de la medida con la función no lineal $h(x,t)$ y la predicción de la medida con el modelo linealizado \mathbf{H}_{t+1} . La no linealidad de la función $h(x,t)$ también obliga en este caso a realizar la conversión de la variable $\hat{\mathbf{y}}_{t+1/t}$ en el espacio de información a la variable $\hat{\mathbf{x}}_{t+1/t} = \mathbf{Y}_{t+1/t}^{-1}\hat{\mathbf{y}}_{t+1/t}$ en el espacio de estado.

▪ **Sistemas no lineales multisensoriales**

Este sistema combina las características de los sistemas multisensoriales lineales con las de los sistemas no lineales monosensoriales, por lo que las expresiones de la etapa de asimilación de los filtros Extendidos de Kalman y de Información para este caso combinan las características expuestas en los dos casos anteriores. Las ecuaciones de asimilación del EKF y del EIF para este tipo de sistemas se recogen en la tabla A1-10.

A1.2.3. Validación

Por último, la etapa opcional de validación, en la que se decide si la medida debe de ser asimilada o desechada, se realiza en un nuevo espacio: el de medida. Aunque dicho espacio de trabajo se encuentra relacionado con el espacio de estado y el espacio de información sobre el estado, la relación es más sencilla si se establece, a partir de las ecuaciones que implementan el modelo de medida, con el espacio de estado. Por lo tanto, utilizaremos la misma validación en el caso de los filtros de Kalman que en el caso de los de Información, lo que resulta ventajoso para todas las versiones del filtro de Kalman ya que el cálculo de la $\hat{\mathbf{r}}_{t+1/t}$ y $\mathbf{S}_{t+1/t}$ será utilizado en la etapa de asimilación. Las expresiones de la etapa de validación para cada tipo de sistema se encuentran recogidas en las tablas A1-7, A1-8, A1-9 y A1-10.

A1.2.4. Tablas con las ecuaciones de los filtros de Kalman y de Información asociados a cada tipo de sistema.

Etapas	Filtro de Kalman	Filtro de Información
Predicción	$\hat{\mathbf{x}}_{t+1/t} = \mathbf{F}_t \hat{\mathbf{x}}_{t/t} + \mathbf{u}_t$ $\mathbf{P}_{t+1/t} = \mathbf{F}_t \mathbf{P}_{t/t} \mathbf{F}_t^T + \mathbf{Q}_t$	$\hat{\mathbf{y}}_{t+1/t} = \mathbf{Y}_{t+1/t} (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \hat{\mathbf{y}}_{t/t} + \mathbf{u}_t)$ $\mathbf{Y}_{t+1/t} = (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \mathbf{F}_t^T + \mathbf{Q}_t)^{-1}$
Validación (opcional)	$\hat{\mathbf{r}}_{t+1/t} = \zeta_{t+1} - \mathbf{H}_{t+1} \hat{\mathbf{x}}_{t+1/t}$ $\mathbf{S}_{t+1/t} = \mathbf{H}_{t+1} \mathbf{P}_{t+1/t} \mathbf{H}_{t+1}^T + \mathbf{R}_{t+1}$ Test: $\hat{\mathbf{r}}_{t+1/t}^T \mathbf{S}_{t+1/t}^{-1} \hat{\mathbf{r}}_{t+1/t} \leq \gamma$?	$\hat{\mathbf{r}}_{t+1/t} = \zeta_{t+1} - \mathbf{H}_{t+1} \mathbf{Y}_{t+1/t}^{-1} \hat{\mathbf{y}}_{t+1/t}$ $\mathbf{S}_{t+1/t} = \mathbf{H}_{t+1} \mathbf{Y}_{t+1/t}^{-1} \mathbf{H}_{t+1}^T + \mathbf{R}_{t+1}$ Test: $\hat{\mathbf{r}}_{t+1/t}^T \mathbf{S}_{t+1/t}^{-1} \hat{\mathbf{r}}_{t+1/t} \leq \gamma$?
Asimilación	$\hat{\mathbf{r}}_{t+1/t} = \zeta_{t+1} - \mathbf{H}_{t+1} \hat{\mathbf{x}}_{t+1/t}$ $\mathbf{S}_{t+1/t} = \mathbf{H}_{t+1} \mathbf{P}_{t+1/t} \mathbf{H}_{t+1}^T + \mathbf{R}_{t+1}$ $\mathbf{W}_{t+1/t} = \mathbf{P}_{t+1/t} \mathbf{H}_{t+1}^T \mathbf{S}_{t+1/t}^{-1}$ $\hat{\mathbf{x}}_{t+1/t+1} = \hat{\mathbf{x}}_{t+1/t} + \mathbf{W}_{t+1/t} \hat{\mathbf{r}}_{t+1/t}$ $\mathbf{P}_{t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1/t} \mathbf{S}_{t+1/t} \mathbf{W}_{t+1/t}^T$	$\mathbf{i}_{t+1} = \mathbf{H}_{t+1}^T \mathbf{R}_{t+1}^{-1} \zeta_{t+1}$ $\mathbf{I}_{t+1} = \mathbf{H}_{t+1}^T \mathbf{R}_{t+1}^{-1} \mathbf{H}_{t+1}$ $\hat{\mathbf{y}}_{t+1/t+1} = \hat{\mathbf{y}}_{t+1/t} + \mathbf{i}_{t+1}$ $\mathbf{Y}_{t+1/t+1} = \mathbf{Y}_{t+1/t} + \mathbf{I}_{t+1}$

Tabla A1-7. Ecuaciones del KF y del IF para un sistema lineal dinámico con un único sensor.

Etapas	Filtro de Kalman	Filtro de Información
Predicción	$\hat{\mathbf{x}}_{t+1/t} = \mathbf{F}_t \hat{\mathbf{x}}_{t/t} + \mathbf{u}_t$ $\mathbf{P}_{t+1/t} = \mathbf{F}_t \mathbf{P}_{t/t} \mathbf{F}_t^T + \mathbf{Q}_t$	$\hat{\mathbf{y}}_{t+1/t} = \mathbf{Y}_{t+1/t} (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \hat{\mathbf{y}}_{t/t} + \mathbf{u}_t)$ $\mathbf{Y}_{t+1/t} = (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \mathbf{F}_t^T + \mathbf{Q}_t)^{-1}$
Validación (opcional)	$\hat{\mathbf{r}}_{s,t+1/t} = \zeta_{s,t+1} - \mathbf{H}_{s,t+1} \hat{\mathbf{x}}_{s,t+1/t}$ $\mathbf{S}_{s,t+1/t} = \mathbf{H}_{s,t+1} \mathbf{P}_{s,t+1/t} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ Test: $\zeta_{s,t+1} \hat{\mathbf{r}}_{s,t+1/t}^T \mathbf{S}_{s,t+1/t}^{-1} \hat{\mathbf{r}}_{s,t+1/t} \leq \gamma_s$?	$\hat{\mathbf{r}}_{s,t+1/t} = \zeta_{s,t+1} - \mathbf{H}_{s,t+1} \mathbf{Y}_{t+1/t}^{-1} \hat{\mathbf{y}}_{t+1/t}$ $\mathbf{S}_{s,t+1/t} = \mathbf{H}_{s,t+1} \mathbf{Y}_{t+1/t}^{-1} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ Test: $\zeta_{s,t+1} \hat{\mathbf{r}}_{s,t+1/t}^T \mathbf{S}_{s,t+1/t}^{-1} \hat{\mathbf{r}}_{s,t+1/t} \leq \gamma_s$?
Asimilación	$\hat{\mathbf{r}}_{s,t+1/t} = \zeta_{s,t+1} - \mathbf{H}_{s,t+1} \hat{\mathbf{x}}_{s,t+1/t}$ $\mathbf{S}_{s,t+1/t} = \mathbf{H}_{s,t+1} \mathbf{P}_{s,t+1/t} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ $\mathbf{W}_{s,t+1/t} = \mathbf{P}_{s,t+1/t} \mathbf{H}_{s,t+1}^T \mathbf{S}_{s,t+1/t}^{-1}$ $\tilde{\mathbf{x}}_{s,t+1/t+1} = \hat{\mathbf{x}}_{t+1/t} + \mathbf{W}_{s,t+1/t} \hat{\mathbf{r}}_{s,t+1/t}$ $\tilde{\mathbf{P}}_{s,t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{s,t+1/t} \mathbf{S}_{s,t+1/t} \mathbf{W}_{s,t+1/t}^T$ $\mathbf{c}_x = \sum_{s=1}^S (\tilde{\mathbf{P}}_{s,t+1/t+1}^{-1} \tilde{\mathbf{x}}_{s,t+1/t+1} - \mathbf{P}_{t+1/t}^{-1} \hat{\mathbf{x}}_{t+1/t})$ $\hat{\mathbf{x}}_{t+1/t+1} = \mathbf{P}_{t+1/t+1} (\mathbf{P}_{t+1/t}^{-1} \hat{\mathbf{x}}_{t+1/t} + \mathbf{c}_x)$ $\mathbf{P}_{t+1/t+1} = (\mathbf{P}_{t+1/t}^{-1} + \sum_{s=1}^S (\tilde{\mathbf{P}}_{s,t+1/t+1}^{-1} - \mathbf{P}_{t+1/t}^{-1}))^{-1}$	$\mathbf{i}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}$ $\mathbf{I}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1}$ $\hat{\mathbf{y}}_{t+1/t+1} = \hat{\mathbf{y}}_{t+1/t} + \sum_{s=1}^S \mathbf{i}_{s,t+1}$ $\mathbf{Y}_{t+1/t+1} = \mathbf{Y}_{t+1/t} + \sum_{s=1}^S \mathbf{I}_{s,t+1}$

Tabla A1-8. Ecuaciones del KF y del IF para un sistema lineal dinámico multisensorial.

Etapas	Filtro de Kalman Extendido	Filtro de Información Extendido
Predicción	$\mathbf{F}_t = (\mathbf{J}f)_{\mathbf{x}}(\hat{\mathbf{x}}_{t/t}, \mathbf{u}_t, t)$ $\hat{\mathbf{x}}_{t+1/t} = f(\hat{\mathbf{x}}_{t/t}, \mathbf{u}_t, t)$ $\mathbf{P}_{t+1/t} = \mathbf{F}_t \mathbf{P}_{t/t} \mathbf{F}_t^T + \mathbf{Q}_t$	$\hat{\mathbf{x}}_{t/t} = \mathbf{Y}_{t/t}^{-1} \hat{\mathbf{y}}_{t/t}$ $\mathbf{F}_t = (\mathbf{J}f)_{\mathbf{x}}(\hat{\mathbf{x}}_{t/t}, \mathbf{u}_t, t)$ $\hat{\mathbf{y}}_{t+1/t} = \mathbf{Y}_{t+1/t} f(\hat{\mathbf{x}}_{t/t}, \mathbf{u}_t, t)$ $\mathbf{Y}_{t+1/t} = (\mathbf{F}_t \mathbf{Y}_{t/t}^{-1} \mathbf{F}_t^T + \mathbf{Q}_t)^{-1}$
Validación (opcional)	$\mathbf{H}_{t+1} = (\mathbf{J}h)_{\mathbf{x}}(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\hat{\mathbf{r}}_{t+1/t} = \zeta_{t+1} - h(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\mathbf{S}_{t+1/t} = \mathbf{H}_{t+1} \mathbf{P}_{t+1/t} \mathbf{H}_{t+1}^T + \mathbf{R}_{t+1}$ Test: $\zeta_{t+1} \hat{\mathbf{r}}_{t+1/t}^T \mathbf{S}_{t+1/t}^{-1} \hat{\mathbf{r}}_{t+1/t} \leq \gamma$?	$\hat{\mathbf{x}}_{t+1/t} = \mathbf{Y}_{t+1/t}^{-1} \hat{\mathbf{y}}_{t+1/t}$ $\mathbf{H}_{t+1} = (\mathbf{J}h)_{\mathbf{x}}(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\hat{\mathbf{r}}_{t+1/t} = \zeta_{t+1} - h(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\mathbf{S}_{t+1/t} = \mathbf{H}_{t+1} \mathbf{Y}_{t+1/t}^{-1} \mathbf{H}_{t+1}^T + \mathbf{R}_{t+1}$ Test: $\zeta_{t+1} \hat{\mathbf{r}}_{t+1/t}^T \mathbf{S}_{t+1/t}^{-1} \hat{\mathbf{r}}_{t+1/t} \leq \gamma$?
Asimilación	$\mathbf{H}_{t+1} = (\mathbf{J}h)_{\mathbf{x}}(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\hat{\mathbf{r}}_{t+1/t} = \zeta_{t+1} - h(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\mathbf{S}_{t+1/t} = \mathbf{H}_{t+1} \mathbf{P}_{t+1/t} \mathbf{H}_{t+1}^T + \mathbf{R}_{t+1}$ $\mathbf{W}_{t+1/t} = \mathbf{P}_{t+1/t} \mathbf{H}_{t+1}^T \mathbf{S}_{t+1/t}^{-1}$ $\hat{\mathbf{x}}_{t+1/t+1} = \hat{\mathbf{x}}_{t+1/t} + \mathbf{W}_{t+1/t} \hat{\mathbf{r}}_{t+1/t}$ $\mathbf{P}_{t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1/t} \mathbf{S}_{t+1/t} \mathbf{W}_{t+1/t}^T$	$\hat{\mathbf{x}}_{t+1/t} = \mathbf{Y}_{t+1/t}^{-1} \hat{\mathbf{y}}_{t+1/t}$ $\mathbf{H}_{t+1} = (\mathbf{J}h)_{\mathbf{x}}(\hat{\mathbf{x}}_{t+1/t}, t+1)$ $\zeta_{t+1}^C = \zeta_{t+1} - h(\hat{\mathbf{x}}_{t+1/t}, t+1) + \mathbf{H}_{t+1} \hat{\mathbf{x}}_{t+1/t}$ $\mathbf{i}_{t+1} = \mathbf{H}_{t+1}^T \mathbf{R}_{t+1}^{-1} \zeta_{t+1}^C$ $\mathbf{I}_{t+1} = \mathbf{H}_{t+1}^T \mathbf{R}_{t+1}^{-1} \mathbf{H}_{t+1}$ $\hat{\mathbf{y}}_{t+1/t+1} = \hat{\mathbf{y}}_{t+1/t} + \mathbf{i}_{t+1}$ $\mathbf{Y}_{t+1/t+1} = \mathbf{Y}_{t+1/t} + \mathbf{I}_{t+1}$

Tabla A1-9. Ecuaciones del EKF y del EIF para un sistema no lineal dinámico con un único sensor.

Etapas	Filtro de Kalman Extendido	Filtro de Información Extendido
Predicción	$\mathbf{F}_t = (\mathbf{J}f)_x(\hat{\mathbf{x}}_{t t}, \mathbf{u}_t, t)$ $\hat{\mathbf{x}}_{t+1 t} = f(\hat{\mathbf{x}}_{t t}, \mathbf{u}_t, t)$ $\mathbf{P}_{t+1 t} = \mathbf{F}_t \mathbf{P}_{t t} \mathbf{F}_t^T + \mathbf{Q}_t$	$\hat{\mathbf{x}}_{t t} = \mathbf{Y}_{t t}^{-1} \hat{\mathbf{y}}_{t t}$ $\mathbf{F}_t = (\mathbf{J}f)_x(\hat{\mathbf{x}}_{t t}, \mathbf{u}_t, t)$ $\hat{\mathbf{y}}_{t+1 t} = \mathbf{Y}_{t+1 t} f(\hat{\mathbf{x}}_{t t}, \mathbf{u}_t, t)$ $\mathbf{Y}_{t+1 t} = (\mathbf{F}_t \mathbf{Y}_{t t} \mathbf{F}_t^T + \mathbf{Q}_t)^{-1}$
Validación (opcional)	$\mathbf{H}_{s,t+1} = (\mathbf{J}h_s)_x(\hat{\mathbf{x}}_{t+1 t}, t+1)$ $\hat{\mathbf{r}}_{s,t+1 t} = \zeta_{s,t+1} - h_s(\hat{\mathbf{x}}_{t+1 t}, t+1)$ $\mathbf{S}_{s,t+1 t} = \mathbf{H}_{s,t+1} \mathbf{P}_{t+1 t} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ $\text{Test: } \hat{\mathbf{r}}_{s,t+1 t}^T \mathbf{S}_{s,t+1 t}^{-1} \hat{\mathbf{r}}_{s,t+1 t} \leq \gamma_s \quad ?$	$\hat{\mathbf{x}}_{t+1 t} = \mathbf{Y}_{t+1 t}^{-1} \hat{\mathbf{y}}_{t+1 t}$ $\mathbf{H}_{s,t+1} = (\mathbf{J}h_s)_x(\hat{\mathbf{x}}_{t+1 t}, t+1)$ $\hat{\mathbf{r}}_{s,t+1 t} = \zeta_{s,t+1} - h_s(\hat{\mathbf{x}}_{t+1 t}, t+1)$ $\mathbf{S}_{s,t+1 t} = \mathbf{H}_{s,t+1} \mathbf{Y}_{t+1 t}^{-1} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ $\text{Test: } \hat{\mathbf{r}}_{s,t+1 t}^T \mathbf{S}_{s,t+1 t}^{-1} \hat{\mathbf{r}}_{s,t+1 t} \leq \gamma_s \quad ?$
Asimilación	$\mathbf{H}_{s,t+1} = (\mathbf{J}h_s)_x(\hat{\mathbf{x}}_{t+1 t}, t+1)$ $\hat{\mathbf{r}}_{s,t+1 t} = \zeta_{s,t+1} - h_s(\hat{\mathbf{x}}_{t+1 t}, t+1)$ $\mathbf{S}_{s,t+1 t} = \mathbf{H}_{s,t+1} \mathbf{P}_{t+1 t} \mathbf{H}_{s,t+1}^T + \mathbf{R}_{s,t+1}$ $\mathbf{W}_{s,t+1 t} = \mathbf{P}_{t+1 t} \mathbf{H}_{s,t+1}^T \mathbf{S}_{s,t+1 t}^{-1}$ $\tilde{\mathbf{x}}_{s,t+1 t+1} = \hat{\mathbf{x}}_{t+1 t} + \mathbf{W}_{s,t+1 t} \hat{\mathbf{r}}_{s,t+1 t}$ $\tilde{\mathbf{P}}_{s,t+1 t+1} = \mathbf{P}_{t+1 t} - \mathbf{W}_{s,t+1 t} \mathbf{S}_{s,t+1 t} \mathbf{W}_{s,t+1 t}^T$ $\mathbf{c}_x = \sum_{s=1}^S (\tilde{\mathbf{P}}_{s,t+1 t+1}^{-1} \tilde{\mathbf{x}}_{s,t+1 t+1} - \mathbf{P}_{t+1 t}^{-1} \hat{\mathbf{x}}_{t+1 t})$ $\hat{\mathbf{x}}_{t+1 t+1} = \mathbf{P}_{t+1 t+1} (\mathbf{P}_{t+1 t}^{-1} \hat{\mathbf{x}}_{t+1 t} + \mathbf{c}_x)$ $\mathbf{P}_{t+1 t+1} = (\mathbf{P}_{t+1 t}^{-1} + \sum_{s=1}^S (\tilde{\mathbf{P}}_{s,t+1 t+1}^{-1} - \mathbf{P}_{t+1 t}^{-1}))^{-1}$	$\hat{\mathbf{x}}_{t+1 t} = \mathbf{Y}_{t+1 t}^{-1} \hat{\mathbf{y}}_{t+1 t}$ $\mathbf{H}_{s,t+1} = (\mathbf{J}h_s)_x(\hat{\mathbf{x}}_{t+1 t}, t+1)$ $\zeta_{s,t+1}^C = \zeta_{s,t+1} - h_s(\hat{\mathbf{x}}_{t+1 t}, t+1) + \mathbf{H}_{s,t+1} \hat{\mathbf{x}}_{t+1 t}$ $\mathbf{i}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}^C$ $\mathbf{I}_{s,t+1} = \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1}$ $\hat{\mathbf{y}}_{t+1 t+1} = \hat{\mathbf{y}}_{t+1 t} + \sum_{s=1}^S \mathbf{i}_{s,t+1}$ $\mathbf{Y}_{t+1 t+1} = \mathbf{Y}_{t+1 t} + \sum_{s=1}^S \mathbf{I}_{s,t+1}$

Tabla A1-10. Ecuaciones del EKF y del EIF para un sistema no lineal dinámico multisensorial.

Apéndice II

Algoritmos propuestos para estimar el estado de sistemas de control con sensores asíncronos y medidas retardadas y desordenadas

En este apéndice se recopilan, a modo de resumen, los principales algoritmos propuestos en esta tesis para estimar el estado de un sistema a partir de la información proporcionada por un conjunto de sensores asíncronos cuyas medidas son recibidas por los algoritmos de forma temporalmente desordenada y/o retardada.

Para todos los algoritmos propuestos, el comportamiento del sistema es modelado mediante redes bayesianas que comparten la misma estructura. El tipo de relaciones de dependencia capturado por la red es la que determina los algoritmos que se pueden utilizar en cada caso.

Cuando el comportamiento del sistema es modelable con funciones de transición de estado y de medida lineales, que incluyen términos aditivos de ruido gaussiano, se pueden utilizar todos los algoritmos propuestos en los capítulos II y los dos primeros del capítulo III. En este apéndice resumen únicamente se recopilan los del capítulo III, ya que la funcionalidad de todos los algoritmos propuestos en el capítulo II se puede englobar en el primer algoritmo del capítulo III. Los dos algoritmos elegidos para este tipo de sistemas obtienen los mismos resultados, salvo errores de redondeo, diferenciándose en el método elegido para estimar el estado del sistema. El primero, propuesto en el capítulo III bajo el epígrafe algoritmo C3-1, utiliza una versión optimizada, para la red bayesiana elegida, del algoritmo de inferencia exacto basado en el árbol de unión para redes bayesiana gaussianas. El segundo, presentado en el mismo capítulo bajo el epígrafe algoritmo C3-2, utiliza las operaciones habituales del filtro de Información en el orden apropiado para estimar el estado del sistema a partir de medidas temporalmente retardadas y desordenadas. Las diferencias de cálculo y de memoria existentes entre los dos algoritmos se encuentran analizadas en el capítulo III. En la primera sección del capítulo IV se comparan los resultados obtenidos por ambos algoritmos para un conjunto de datos experimentales generados sobre un único sistema ejemplo.

Para sistemas modelables con funciones de transición de estado y de medida lineales y/o no lineales, que incorporan términos aditivos de ruido gaussiano, en el capítulo III se propone un tercer algoritmo que es capaz de trabajar, de forma adecuada, con la falta de linealidad de alguna de las ecuaciones que modelan el comportamiento del sistema. Este algoritmo, propuesto en el capítulo III bajo el epígrafe algoritmo C3-3 es soportado sobre las operaciones del filtro de Información y del filtro de Información Extendido para las ecuaciones del sistema lineales y no lineales respectivamente. También puede ser utilizado para sistemas gaussianos puramente lineales, ya que

su funcionalidad engloba la del segundo algoritmo propuesto en el capítulo III. Su comportamiento es analizado en la segunda sección del capítulo IV sobre diferentes sistemas ejemplo.

Además de los tres algoritmos propuestos en el capítulo III, en este apéndice se presenta un algoritmo adicional, que constituye una versión modificada del algoritmo C3-3 que reduce el número de operaciones, eliminando el recálculo de la información sensorial asociada a los sensores no lineales, a costa de cometer un error en los valores estimados por el algoritmo. La importancia de dicho error dependerá de la influencia que la no linealidad del sistema tiene sobre la reasimilación de las medidas de los sensores no lineales. Este hecho se analiza en el capítulo IV sobre diferentes sistemas ejemplo.

Los cuatro algoritmos recogidos en éste apéndice se encuentran implementados en Matlab (versión 5.3 y posteriores) bajo la forma de una herramienta (librería) fácilmente configurable, que ha sido utilizada durante el estudio del funcionamiento exhaustivo de los algoritmos del capítulo IV. Dicha herramienta permite estimar el estado de cualquier sistema modelable con funciones de transición de estado y de medida que incluyen términos aditivos de ruido gaussiano, tras la definición del conjunto de funciones de Matlab que implementan las ecuaciones ideales del sistema y de medida, y la especificación de las matrices de covarianza asociadas al ruido.

Además, los algoritmos C3-2 y la versión original del algoritmo C3-3 se encuentran implementados en una librería C++ fácilmente adaptable a diferentes sistemas a través de la definición de un conjunto de clases que recogen las características propias del sistema modelado (ecuaciones de transición de estado y de medida del sistema, y las matrices de covarianza asociadas al término aditivo de ruido gaussiano). La librería se ha utilizado sobre los ejemplos del capítulo IV asociados al algoritmo C3-2 y C3-3 original, obteniéndose resultados semejantes, aunque no idénticos debido a las diferencias de redondeo introducidas por Matlab y el programa ejecutable obtenido a partir de la librería C++. En el capítulo V se utiliza una versión adaptada de la librería para estimar la posición de un robot real. Las características del algoritmo implementado en la versión adaptada, y el porqué de los cambios realizados, se describen en el apéndice V.

Por último es importante destacar que para facilitar la comparativa de cada pareja de algoritmos, formada por los algoritmos C3-1 y C3-2 para sistemas gaussianos puramente lineales y por los algoritmos C3-3 original y modificado para sistemas gaussianos mixtos, cada pareja es recogida en una columna de una única tabla, dividida en un conjunto de filas que permiten agrupar las operaciones de las diferentes etapas del algoritmo.

A2.1. Tablas resumen con los algoritmos propuestos

Las expresiones de los algoritmos propuestos en el capítulo III bajo los epígrafes C3-1 y C3-2 para estimar el estado de sistemas de control con modelos de transición de estado y de medida lineales se recogen en la tabla A2-1, mientras que los algoritmos propuestos para estimar el estado de sistema con ecuaciones de transición o medida no lineales se exponen en la tabla A2-2. Los algoritmos han sido renombrados de acuerdo con la nomenclatura elegida al inicio del capítulo IV: Bayes es el algoritmo C3-1, IFAsyn el algoritmo C3-2, IFAsynMixto la versión original del algoritmo C3-3 e IFAsynMixtoEsp la versión modificada del algoritmo C3-3.

La estructura de ambas tablas es la siguiente. En las primeras columnas, cuyo texto aparece orientado de forma diferente que en las dos últimas, se indica en que etapa se encuentran los dos algoritmos recogidos en cada tabla. En las dos últimas columnas, bajo los encabezados asociados a cada uno de los dos algoritmos de la tabla, se encuentran las operaciones realizadas dentro de la etapa especificada en las columnas ortogonales de la izquierda, para cada algoritmo. En los casos en los que las operaciones realizadas por ambos algoritmos sean iguales, éstas se exponen en una columna común a los dos algoritmos, formada a partir de la eliminación de la división existente entre las columnas asociadas a cada algoritmo.

Por último es importante destacar que la versión modificada del algoritmo C3-3, únicamente se encuentra recogida en este apéndice, ya que en principio fue solamente desarrollada para analizar en el capítulo IV la necesidad del recálculo de la información sensorial asociada a los sensores no lineales cuando el algoritmo recibía medidas temporalmente retardadas. En la versión modificada, incluida en este apéndice para permitir que el lector la compare con la versión original, se elimina dicho recálculo, hecho que reduce considerablemente las necesidades de cómputo y de memoria del algoritmo. Para medir la necesidad de dicho recálculo, se ejecutarán en el capítulo IV las dos versiones del algoritmo C3-3 (IFAsynMixto e IFAsynMixtoEsp) sobre los mismos datos experimentales, y se compararán los resultados obtenidos.

Apéndice II: Algoritmos propuestos para estimar el estado de sistemas de control con sensores asíncronos y medidas retardadas y desordenadas

	Bayes	IFAsyn
Inicialización	<p>a. Obtener el modelo del sistema (ec. de transición de estados) y de los sensores (ec. de medida).</p> <p>b. Fijar el tamaño de la ventana de tiempo n.</p> <p>c. $t=1$</p> <p>d. Construir el árbol de unión inicial, formado únicamente por el subárbol asociado al instante de muestro $t=1$ (es decir, los conglomerados $\mathbf{x}_0\mathbf{x}_1$, y $\mathbf{x}_1\mathbf{z}_{s,1}$, y los separadores $\mathbf{x}_1\mathbf{z}_{s,1} \cap \mathbf{x}_0\mathbf{x}_1$ y $\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{z}_{s,1}$).</p> <p>e. Inicializar el valor de los potenciales de los elementos iniciales:</p> $\Phi_{\mathbf{x}_0\mathbf{x}_1}(\mathbf{x}_0, \mathbf{x}_1) = \left(g, \begin{bmatrix} \hat{\mathbf{y}}_0 - \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{u}_0 \\ \mathbf{Q}_0^{-1} \mathbf{u}_0 \end{bmatrix}, \begin{bmatrix} \hat{\mathbf{Y}}_0 + \mathbf{F}_0^T \mathbf{Q}_0^{-1} \mathbf{F}_0 & -\mathbf{F}_0^T \mathbf{Q}_0^{-1} \\ -\mathbf{Q}_0^{-1} \mathbf{F}_0 & \mathbf{Q}_0^{-1} \end{bmatrix} \right)$ $\Phi_{\mathbf{x}_1\mathbf{z}_{s,1}}(\mathbf{x}_1, \mathbf{z}_{s,1}) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,1}^T \mathbf{R}_{s,1}^{-1} \mathbf{H}_{s,1} & -\mathbf{H}_{s,1}^T \mathbf{R}_{s,1}^{-1} \\ -\mathbf{R}_{s,1}^{-1} \mathbf{H}_{s,1} & \mathbf{R}_{s,1}^{-1} \end{bmatrix} \right)$ $\Phi_{\mathbf{x}_1\mathbf{z}_{s,1} \cap \mathbf{x}_0\mathbf{x}_1}(\mathbf{x}_1) = (g, [\mathbf{0}], [\mathbf{0}]) \quad \Phi_{\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{z}_{s,1}}(\mathbf{x}_1) = (g, [\mathbf{0}], [\mathbf{0}])$ <p>f. Actualizar el valor del potencial del separador $\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{z}_{s,1}$ a partir de una marginalización sobre el potencial del conglomerado $\mathbf{x}_0\mathbf{x}_1$</p> $\Phi_{\mathbf{x}_0\mathbf{x}_1 \cap \mathbf{x}_1\mathbf{z}_{s,1}}(\mathbf{x}_1) = \int \Phi_{\mathbf{x}_0\mathbf{x}_1}(\mathbf{x}_0, \mathbf{x}_1) d\mathbf{x}_0$	<p>d. Inicializar las variables asociadas al instante $t=1$</p> $\mathbf{Y}_{1 AASIM \leq 0} = (\mathbf{F}_0 \mathbf{P}_0 \mathbf{F}_0^T + \mathbf{Q}_0)^{-1}$ $\hat{\mathbf{y}}_{1 AASIM \leq 0} = \mathbf{Y}_{1 AASIM \leq 0} (\mathbf{F}_0 \hat{\mathbf{x}}_0 + \mathbf{u}_0)$ $\mathbf{i}_1 = \mathbf{0}, \mathbf{I}_1 = \mathbf{0}$
Bucle principal (while true)	<p>a. Ha pasado un periodo de muestreo</p> <p>i. $t=t+1$</p> <p>ii. Añadir los conglomerados y separadores asociados a t (es decir, $\mathbf{x}_{t-1}\mathbf{x}_t$, $\mathbf{x}_t\mathbf{z}_{s,t}$, $\mathbf{x}_t\mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t$ y $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{z}_{s,t}$)</p> <p>iii. Inicializar los potenciales de los elementos añadidos:</p> $\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = \Phi(\mathbf{x}_{t-1}, \mathbf{x}_t) = \left(g, \begin{bmatrix} -\mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \mathbf{u}_{t-1} \\ \mathbf{Q}_{t-1}^{-1} \mathbf{u}_{t-1} \end{bmatrix}, \begin{bmatrix} \mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \mathbf{F}_{t-1} & -\mathbf{F}_{t-1}^T \mathbf{Q}_{t-1}^{-1} \\ -\mathbf{Q}_{t-1}^{-1} \mathbf{F}_{t-1} & \mathbf{Q}_{t-1}^{-1} \end{bmatrix} \right)$ $\Phi_{\mathbf{x}_t\mathbf{z}_{s,t}}(\mathbf{x}_t, \mathbf{z}_{s,t}) = \left(g, \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & -\mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \\ -\mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t} & \mathbf{R}_{s,t}^{-1} \end{bmatrix} \right)$ $\Phi_{\mathbf{x}_t\mathbf{z}_{s,t} \cap \mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_t) = (g, [\mathbf{0}], [\mathbf{0}]) \quad \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{z}_{s,t}}(\mathbf{x}_t) = (g, [\mathbf{0}], [\mathbf{0}])$	<p>i. $t=t+1$</p> <p>ii. Se asimila la información del instante anterior $t-1$</p> $\hat{\mathbf{y}}_{t-1 AASIM \leq t-1} = \hat{\mathbf{y}}_{t-1 AASIM \leq t-2} + \mathbf{i}_{t-1}$ $\mathbf{Y}_{t-1 AASIM \leq t-1} = \mathbf{Y}_{t-1 AASIM \leq t-2} + \mathbf{I}_{t-1}$ <p>iii. Se inicializan las variables para el instante t</p> $\mathbf{Y}_{t AASIM \leq t-1} = (\mathbf{F}_{t-1} \mathbf{Y}_{t-1 AASIM \leq t-1}^{-1} \mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})^{-1}$ $\hat{\mathbf{y}}_{t AASIM \leq t-1} = \mathbf{Y}_{t AASIM \leq t-1} (\mathbf{F}_{t-1} \mathbf{Y}_{t-1 AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t-1 AASIM \leq t-1} + \mathbf{u}_{t-1})$ $\mathbf{i}_t = \mathbf{0}, \mathbf{I}_t = \mathbf{0}$

Tabla A2-1. Algoritmos propuestos para sistemas de control lineales

	Bayes		IFAsyn
Bucle principal (while true)	a. Ha pasado un periodo de muestreo	<p>iv. Incluir en el potencial $\mathbf{x}_{t-1}\mathbf{x}_t$ la información proveniente de instantes interiores:</p> $\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) = \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) \cdot \Phi_{\mathbf{x}_{t-2}\mathbf{x}_{t-1} \cap \mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1})$ <p>v. Actualizar el valor del potencial $\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}$ a partir de la marginalización de \mathbf{x}_{t-1} sobre el potencial del conglomerado $\mathbf{x}_{t-1}\mathbf{x}_t$:</p> $\Phi_{\mathbf{x}_{t-1}\mathbf{x}_t \cap \mathbf{x}_t\mathbf{x}_{t+1}}(\mathbf{x}_t) = \int \Phi_{\mathbf{x}_{t-1}\mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) d\mathbf{x}_{t-1}$ <p>vi. Si $t-n > 0$ hay que:</p> <ol style="list-style-type: none"> 1. Eliminar los elementos del instante $t-n$, es decir los conglomerados $\mathbf{x}_{t-n-1}\mathbf{x}_{t-n}$, $\mathbf{x}_{t-n}\mathbf{z}_{s,t-n}$, los separadores $\mathbf{x}_{t-n}\mathbf{z}_{s,t-n} \cap \mathbf{x}_{t-n-1}\mathbf{x}_{t-n}$ y $\mathbf{x}_{t-n-1}\mathbf{x}_{t-n} \cap \mathbf{x}_{t-n}\mathbf{x}_{t-n+1}$, y sus potenciales asociados, 2. o reservar la memoria asociada a los mismos para el próximo periodo de muestreo $t+1$. 	<p>iv. Si $t-n > 0$ hay que eliminar los elementos asociados al instante $t-n$, es decir las variables:</p> $\mathbf{Y}_{t-n AASIM \leq t-n-1}, \hat{\mathbf{y}}_{t-n AASIM \leq t-n-1}, \mathbf{i}_{t-n}, \mathbf{I}_{t-n}, \mathbf{u}_{t-n}$
Bucle principal (while true)	b. Hay una nueva medida $\zeta_{s,k}$	<p>i. Si no $(k > t-n)$ $\zeta_{s,k}$ no esta dentro de la ventana de tiempo y no se puede asimilar. Es conveniente mandar un warning para que el usuario se entere</p> <p>ii. Si $(k > t-n)$ $\zeta_{s,k}$ esta dentro de la ventana de tiempo</p> <ol style="list-style-type: none"> 1. Obtener el valor del potencial de transición auxiliar $\Phi_{T_AUX}(\mathbf{x}_{k-1}, \mathbf{x}_k) = \frac{\Phi_{\mathbf{x}_{k-1}\mathbf{x}_k}(\mathbf{x}_{k-1}, \mathbf{x}_k)}{\prod_s \Phi_{\mathbf{x}_k\mathbf{z}_{s,k} \cap \mathbf{x}_{k-1}\mathbf{x}_k}(\mathbf{x}_k)}$ <ol style="list-style-type: none"> 2. Realizar una absorción tipo (4) desde el nodo sensorial asociado a la medida $\Phi_{T_AUX}(\mathbf{x}_{k-1}, \mathbf{x}_k)$ sin actualizar los valores del potencial del separador de medida ni del potencial de medida. Almacenar el resultado en $\Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$ 3. Marginalizar la variable del estado \mathbf{x}_t de $\Phi_{M_AUX}(\mathbf{x}_t, \mathbf{z}_{s,t})$. El resultado de la operación es el potencial $\Phi = (\mathbf{g}, \mathbf{h}, \mathbf{K})$ 4. Obtener el valor estimado de la medida y su covarianza como $\mathbf{K}^{-1}\mathbf{h}$ y \mathbf{K}^{-1} respectivamente 5. Realizar el test de validación externa basado en la estima de la medida y su covarianza (por ejemplo el test chi-cuadrado) 	<ol style="list-style-type: none"> 1. Obtener el valor estimado de la medida $\hat{\mathbf{z}}_{s,k AASIM \leq k-1} = \mathbf{H}_{s,k} \mathbf{Y}_{k AASIM \leq k-1}^{-1} \hat{\mathbf{y}}_{k AASIM \leq k-1}$ $\mathbf{S}_{s,k AASIM \leq k-1} = \mathbf{H}_{s,k} \mathbf{Y}_{k AASIM \leq k-1}^{-1} \mathbf{H}_{s,k}^T + \mathbf{R}_{s,k}$ <ol style="list-style-type: none"> 2. Realizar el test chi-cuadrado de validación basado en el valor estimado de la medida $\hat{\mathbf{r}}_{s,k AASIM \leq k-1} = \zeta_{s,k} - \hat{\mathbf{z}}_{s,k AASIM \leq k-1}$ <p>Test: $\hat{\mathbf{r}}_{s,k AASIM \leq k-1}^T \mathbf{S}_{s,k AASIM \leq k-1}^{-1} \hat{\mathbf{r}}_{s,k AASIM \leq k-1} \leq \gamma_s$?</p>

	Bayes				IFAsyn
Bucle principal (while true)	b. Hay una nueva medida $\zeta_{s,k}$	ii. Si $(k > t - n)$ $\zeta_{s,k}$ esta dentro de la ventana de tiempo	Si la medida es válida o no hay que pasar el test se validación	<p>a. Introducir la medida $\zeta_{k,t}$ como evidencia en el nodo sensorial $\mathbf{x}_k \mathbf{z}_{s,k}$</p> <p>b. Realizar una absorción tipo (3) desde el nodo de transición de estados asociado al nodo sensorial sobre el que se ha medido (absorción $\mathbf{x}_k \mathbf{z}_{s,k} \rightarrow \mathbf{x}_{k-1} \mathbf{x}_k$)</p> <p>c. Si $k \neq t$, realizar absorciones tipo (1) de forma sucesiva y ordenada de izquierda a derecha desde el nodo de transición de estados asociado a la medida disponible hasta el último (primera absorción $\mathbf{x}_{k-1} \mathbf{x}_k \rightarrow \mathbf{x}_k \mathbf{x}_{t+1}$, ... , y última absorción $\mathbf{x}_{t-2} \mathbf{x}_{t-1} \rightarrow \mathbf{x}_{t-1} \mathbf{x}_t$)</p> <p>d. Actualizar el valor del potencial del separador $\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}$ a partir de una marginalización sobre el potencial del conglomerado $\mathbf{x}_{t-1} \mathbf{x}_t$:</p> $\Phi_{\mathbf{x}_{t-1} \mathbf{x}_t \cap \mathbf{x}_t \mathbf{x}_{t+1}}(\mathbf{x}_t) = \int \Phi_{\mathbf{x}_{t-1} \mathbf{x}_t}(\mathbf{x}_{t-1}, \mathbf{x}_t) d\mathbf{x}_{t-1}$	<p>a. Obtener la información sensorial asociada a la medida $\zeta_{s,k}$:</p> $\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \zeta_{s,k}$ $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}$ <p>b. Añadir la nueva información sensorial al resto de la información sensorial asociada a dicho periodo de muestreo:</p> $\mathbf{i}_k = \mathbf{i}_k + \mathbf{i}_{s,k}$ $\mathbf{I}_k = \mathbf{I}_k + \mathbf{I}_{s,k}$ <p>c. Si $k \neq t$, hay que propagar la nueva información sensorial hacia delante, de forma que se actualicen el término aditivo de la etapa de asimilación que incluye la información sobre el estado en instante de muestreo anteriores:</p> <p>for $j = k : t - 1$</p> $\hat{\mathbf{y}}_{j AASIM \leq j} = \hat{\mathbf{y}}_{j AASIM \leq j-1} + \mathbf{i}_j$ $\mathbf{Y}_{j AASIM \leq j} = \mathbf{Y}_{j AASIM \leq j-1} + \mathbf{I}_j$ $\mathbf{Y}_{j+1 AASIM \leq j} = (\mathbf{F}_j \mathbf{Y}_{j AASIM \leq j}^{-1} \mathbf{F}_j^T + \mathbf{Q}_j)^{-1}$ $\hat{\mathbf{y}}_{j+1 AASIM \leq j} = \mathbf{Y}_{j+1 AASIM \leq j} (\mathbf{F}_j \mathbf{Y}_{j AASIM \leq j}^{-1} \hat{\mathbf{y}}_{j AASIM \leq j} + \mathbf{u}_j)$ <p>end</p>

Tabla A2-1. Algoritmos propuestos en el capítulo III de la tesis para estimar el estado de un sistema de control con múltiples sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada, y cuyas funciones de transición de estados y de medida tienen que ser necesariamente lineales.

Tabla A2-2. Algoritmos propuestos para sistemas de control mixtos (lineales y no lineales)

	IFAsynMixto	IFAsynMixtoEsp
Inicialización	<p>a. Obtener el modelo del sistema (ec. de transición de estados) y de los sensores (ec. de medida), e inicializar el vector SLineal (con SLineal(s)=true si s es lineal y SLineal(s)=false) y la matriz SAsimilada a false (SAsimilada(k,s) indicará si el algoritmo ha asimilado la medida del sensor s con en el instante k).</p> <p>b. Fijar el tamaño de la ventana de tiempo n.</p> <p>c. t=1</p> <p>d. Inicializar las variables asociadas al instante t=1</p> $\mathbf{F}_0 = \text{ObtenerF}(\hat{\mathbf{x}}_0, \mathbf{u}_0, 0), \quad \mathbf{Y}_{1/AASIM \leq 0} = (\mathbf{F}_0 \mathbf{P}_0 \mathbf{F}_0^T + \mathbf{Q}_0)^{-1}, \quad \hat{\mathbf{y}}_{1/AASIM \leq 0} = \mathbf{Y}_{1/AASIM \leq 0} \cdot f(\hat{\mathbf{x}}_0, \mathbf{u}_0, 0)$ $\mathbf{iL}_1 = \mathbf{0}, \quad \mathbf{IL}_1 = \mathbf{0}, \quad \mathbf{iNL}_1 = \mathbf{0}, \quad \mathbf{INL}_1 = \mathbf{0}$	
Bucle principal (while true)	a. Ha pasado el tiempo de un muestreo	<p>Hay que inicializar las variables del nuevo instante de muestreo, y posiblemente eliminar las del periodo más antiguo</p> <p>i. t=t+1</p> <p>ii. Se asimila la información del instante anterior t-1</p> $\hat{\mathbf{y}}_{t-1/AASIM \leq t-1} = \hat{\mathbf{y}}_{t-1/AASIM \leq t-2} + \mathbf{iL}_{t-1} + \mathbf{iNL}_{t-1}, \quad \mathbf{Y}_{t-1/AASIM \leq t-1} = \mathbf{Y}_{t-1/AASIM \leq t-2} + \mathbf{IL}_{t-1} + \mathbf{INL}_{t-1}$ <p>iii. Se inicializan las variables para el instante t</p> $\hat{\mathbf{x}}_{t-1/AASIM \leq t-1} = \mathbf{Y}_{t-1/AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t-1/AASIM \leq t-1}, \quad \mathbf{F}_{t-1} = \text{ObtenerF}(\hat{\mathbf{x}}_{t-1/AASIM \leq t-1}, \mathbf{u}_{t-1}, t-1)$ $\mathbf{Y}_{t/AASIM \leq t-1} = (\mathbf{F}_{t-1} \mathbf{Y}_{t-1/AASIM \leq t-1}^{-1} \mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})^{-1}, \quad \hat{\mathbf{y}}_{t/AASIM \leq t-1} = \mathbf{Y}_{t/AASIM \leq t-1} \cdot f(\hat{\mathbf{x}}_{t-1/AASIM \leq t-1}, \mathbf{u}_{t-1}, t-1)$ $\mathbf{iL}_t = \mathbf{0}, \quad \mathbf{IL}_t = \mathbf{0}, \quad \mathbf{iNL}_t = \mathbf{0}, \quad \mathbf{INL}_t = \mathbf{0}$ <p>iv. Si t-n>0 hay que eliminar los elementos asociados al instante t-n, es decir las variables:</p> $\mathbf{Y}_{t-n/AASIM \leq t-n-1}, \quad \hat{\mathbf{y}}_{t-n/AASIM \leq t-n-1}, \quad \mathbf{u}_{t-n}, \quad \mathbf{iL}_{t-n}, \quad \mathbf{IL}_{t-n}, \quad \mathbf{iNL}_{t-n}, \quad \mathbf{INL}_{t-n}$ <p>$\zeta_{s,t-n}$ para los s no lineales</p>
		i. Si no (k>t-n) $\zeta_{s,k}$ no esta dentro de la ventana de tiempo y no se puede asimilar
	b. Hay una nueva medida $\zeta_{s,k}$	<p>1. Obtener el valor estimado de la medida (cuando haya que hacer el test de validación)</p> $\hat{\mathbf{x}}_{k/AASIM \leq k-1} = \mathbf{Y}_{k/AASIM \leq k-1}^{-1} \hat{\mathbf{y}}_{k/AASIM \leq k-1}, \quad \mathbf{H}_{s,k} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k),$ $\hat{\mathbf{z}}_{s,k/AASIM \leq k-1} = h_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k) \mathbf{S}_{s,k/AASIM \leq k-1} = \mathbf{H}_{s,k} \mathbf{Y}_{k/AASIM \leq k-1}^{-1} \mathbf{H}_{s,k}^T + \mathbf{R}_{s,k}$ <p>2. Realizar el test chi-cuadrado de validación basado en el valor estimado de la medida (cuando haya que hacer el test de validación. Si el test no es necesario la medida es válida)</p> $\hat{\mathbf{r}}_{s,k/AASIM \leq k-1} = \zeta_{s,k} - \hat{\mathbf{z}}_{s,k/AASIM \leq k-1}, \quad \text{Test: } \hat{\mathbf{r}}_{s,k/AASIM \leq k-1}^T \mathbf{S}_{s,k/AASIM \leq k-1}^{-1} \hat{\mathbf{r}}_{s,k/AASIM \leq k-1} \leq \gamma_s \quad ?$
	ii. Si (k>t-n) $\zeta_{s,k}$ esta dentro de la ventana de tiempo	<p>3. Si $\zeta_{s,k}$ es válida</p> <p>a. Obtener la información sensorial asociada a la medida (en b.ii.1 ya se ha obtenido la $\mathbf{H}_{s,k}$ y $\hat{\mathbf{x}}_{k/AASIM \leq k-1}$) e indicar que va a ser asimilada</p> <p>if SLineal(s)=true, $\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \zeta_{s,k}$, $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}$</p> <p>else</p> $\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} (\zeta_{s,k} - h_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k) + \mathbf{H}_{s,k} \hat{\mathbf{x}}_{t/AASIM \leq k-1}),$ $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}, \quad \text{Almacenar } \zeta_{s,k}$ <p>end</p> <p>SAsimilada(k,s)=true</p> <p>b. Añadir la nueva información sensorial al resto de la información sensorial, del mismo tipo, asociada a dicho periodo de muestreo:</p> <p>if SLineal(s)=1 $\mathbf{iL}_k = \mathbf{iL}_k + \mathbf{i}_{s,k}$, $\mathbf{IL}_k = \mathbf{IL}_k + \mathbf{I}_{s,k}$</p> <p>else $\mathbf{iNL}_k = \mathbf{iNL}_k + \mathbf{i}_{s,k}$, $\mathbf{INL}_k = \mathbf{INL}_k + \mathbf{I}_{s,k}$</p> <p>end</p>

Apéndice II: Algoritmos propuestos para estimar el estado de sistemas de control con sensores asíncronos y medidas retardadas y desordenadas

				IFAsynMixto	IFAsynMixtoEsp
Bucle principal (while true)	b. Hay una nueva medida $\zeta_{s,k}$	ii. Si $(k > t - n)$ $\zeta_{s,k}$ esta dentro de la ventana de tiempo	3. Si $\zeta_{s,k}$ es válida	<p>c. Si $k \neq t$, hay que propagar la nueva información sensorial hacia delante, de forma que se actualicen el término aditivo de la etapa de asimilación que incluye la información sobre el estado en instante de muestreo anteriores:</p> <p>Se asimila la información del instante k y se realiza la 1ª predicción</p> $\hat{\mathbf{y}}_{k AASIM \leq k} = \hat{\mathbf{y}}_{k AASIM \leq k-1} + \mathbf{i}L_k + \mathbf{i}NL_k, \quad \mathbf{Y}_{k AASIM \leq k} = \mathbf{Y}_{k AASIM \leq k-1} + \mathbf{I}L_k + \mathbf{I}NL_k$ $\hat{\mathbf{x}}_{k AASIM \leq k} = \mathbf{Y}_{k AASIM \leq k}^{-1} \hat{\mathbf{y}}_{k AASIM \leq k}, \quad \mathbf{F}_k = \text{ObtenerF}(\hat{\mathbf{x}}_{k AASIM \leq k}, \mathbf{u}_k, k)$ $\mathbf{Y}_{k+1 AASIM \leq k} = (\mathbf{F}_k \mathbf{Y}_{k AASIM \leq k}^{-1} \mathbf{F}_k^T + \mathbf{Q}_k)^{-1}$ $\hat{\mathbf{y}}_{k+1 AASIM \leq k} = \mathbf{Y}_{k+1 AASIM \leq k} \cdot f(\hat{\mathbf{x}}_{k AASIM \leq k}, \mathbf{u}_k, k)$ <p>for $j = k+1:t-1$</p>	
				<p>Se recalcula la información sensorial no lineal</p> $\hat{\mathbf{x}}_{j AASIM \leq j-1} = \mathbf{Y}_{j AASIM \leq j-1}^{-1} \hat{\mathbf{y}}_{j AASIM \leq j-1}, \quad \mathbf{i}NL_j = \mathbf{0}, \quad \mathbf{I}NL_j = \mathbf{0}$ <p>for $s = 1:S$</p> <p>if $S\text{Lineal}(s) = \text{false} \wedge S\text{Asimilada}(j, s) = \text{true}$</p> $\mathbf{H}_{s,j} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{j AASIM \leq j-1}, j)$ $\zeta_{s,j}^C = \zeta_{s,j} - h_s(\hat{\mathbf{x}}_{j AASIM \leq j-1}, j) + \mathbf{H}_{s,j} \hat{\mathbf{x}}_{j AASIM \leq j-1}$ $\mathbf{i}_{s,j} = \mathbf{H}_{s,j}^T \mathbf{R}_{s,j}^{-1} \zeta_{s,j}^C, \quad \mathbf{I}_{s,j} = \mathbf{H}_{s,j}^T \mathbf{R}_{s,j}^{-1} \mathbf{H}_{s,j}$ $\mathbf{i}NL_j = \mathbf{i}NL_j + \mathbf{i}_{s,j}, \quad \mathbf{I}NL_j = \mathbf{I}NL_j + \mathbf{I}_{s,j}$ <p>end</p> <p>end</p>	No se recalcula la información sensorial no lineal
				<p>Se asimila la información sensorial para el instante j y se realiza la siguiente predicción</p> $\hat{\mathbf{y}}_{j AASIM \leq j} = \hat{\mathbf{y}}_{j AASIM \leq j-1} + \mathbf{i}L_j + \mathbf{i}NL_j, \quad \mathbf{Y}_{j AASIM \leq j} = \mathbf{Y}_{j AASIM \leq j-1} + \mathbf{I}L_j + \mathbf{I}NL_j$ $\hat{\mathbf{x}}_{j AASIM \leq j} = \mathbf{Y}_{j AASIM \leq j}^{-1} \hat{\mathbf{y}}_{j AASIM \leq j}, \quad \mathbf{F}_j = \text{ObtenerF}(\hat{\mathbf{x}}_{j AASIM \leq j}, \mathbf{u}_j, j)$ $\mathbf{Y}_{j+1 AASIM \leq j} = (\mathbf{F}_j \mathbf{Y}_{j AASIM \leq j}^{-1} \mathbf{F}_j^T + \mathbf{Q}_j)^{-1}$ $\hat{\mathbf{y}}_{j+1 AASIM \leq j} = \mathbf{Y}_{j+1 AASIM \leq j} \cdot f(\hat{\mathbf{x}}_{j AASIM \leq j}, \mathbf{u}_j, j)$ <p>end (del for $j = k+1:t-1$)</p>	
				<p>Se recalcula la información sensorial no lineal del último instante de muestreo</p> $\hat{\mathbf{x}}_{t AASIM \leq t-1} = \mathbf{Y}_{t AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t AASIM \leq t-1}, \quad \mathbf{i}NL_t = \mathbf{0}, \quad \mathbf{I}NL_t = \mathbf{0}$ <p>for $s = 1:S$</p> <p>if $S\text{Lineal}(s) = \text{false} \wedge S\text{Asimilada}(t, s) = \text{true}$</p> $\mathbf{H}_{s,t} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{t AASIM \leq t-1}, t)$ $\zeta_{s,t}^C = \zeta_{s,t} - h_s(\hat{\mathbf{x}}_{t AASIM \leq t-1}, t) + \mathbf{H}_{s,t} \hat{\mathbf{x}}_{t AASIM \leq t-1}$ $\mathbf{i}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \zeta_{s,t}^C, \quad \mathbf{I}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t}$ $\mathbf{i}NL_t = \mathbf{i}NL_t + \mathbf{i}_{s,t}, \quad \mathbf{I}NL_t = \mathbf{I}NL_t + \mathbf{I}_{s,t}$ <p>end</p> <p>end</p>	No se recalcula la información sensorial no lineal

Tabla A2-2. Filtros de Información Mixto, algoritmo C3-3 en su versión original y modificada, para estimar el estado de un sistema de control con múltiples sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada, y cuya función de transición puede ser lineal o no lineal, y cada una de sus funciones de medida lineales o no lineales.

Apéndice III

El Robot Móvil Autónomo Nandana: Descripción Hardware y Software

En este apéndice se describe el robot móvil autónomo Nandana que se ha construido durante el desarrollo de esta tesis en el Departamento de Arquitectura de Computadores y Automática de la Universidad Complutense de Madrid. Sus características vienen determinadas por el principal objetivo perseguido durante la fase de diseño: la creación de una plataforma robótica móvil económica, versátil y abierta, en la que se puedan incorporar fácilmente y de forma progresiva distintos tipos y número de sensores, y que pueda ser utilizada como banco de pruebas para diferentes proyectos de robótica.

Al analizar las plataformas robóticas comerciales existentes, entre las que cabe destacar los populares *Pioneer* de *ActiveMedia Robotics* (www.activrobots.com), los *ATRV* y *B21r* de *IRobot Corporation* (www.irobot.com), el *Robuter* y sucesores *Robular80* y *Robular150* de *Robosoft* (www.robosoft.fr), y los *Scouts* de la ya desaparecida *Nomadic Technologies* (www.robots.com), observamos que se caracterizaban por ser plataformas de elevado coste y cerradas a un conjunto de opciones de configuración hardware (CPUs, elementos de control y unidades sensoriales) y software (sistema operativo y conjunto de librerías/repertorio de instrucciones necesario para el control de los dispositivos hardware del robot) proporcionadas por el fabricante. Por lo tanto, decidimos construir nuestro propio robot partiendo de la sencilla plataforma *Round Max 96* de *Zagros Robotics* (www.zagrosrobotics.com).

El *Round Max 96* está formado por un par de plataformas circulares de 40 cm de diámetro, ensambladas con un grupo de barras verticales con el objeto de constituir el cuerpo básico del robot, y por cuatro ruedas, dos de castor y dos motrices, éstas últimas controlables a través de dos motores DC. Con esta elección, el diseño de toda la arquitectura hardware y software de control es responsabilidad del usuario, que tiene la posibilidad de elegir libremente los elementos que quiera incorporar a su robot, encontrándose únicamente limitado por el peso máximo que éste puede soportar. Como contrapartida a esta libertad de elección, el usuario tiene que responsabilizarse del correcto funcionamiento de toda la electrónica del robot y del control de la misma. Esta labor se simplifica si se utilizan microcontroladores de bajo coste, como son los *MC68000* de *Motorola* (e-www.motorola.com) y los *PIC* de *Microchip* (www.microchip.com), y/o de tarjetas comerciales de adquisición de datos directamente controlables desde un PC, como por ejemplo una tarjeta de adquisición de imágenes *Matrox* (www.matrox.com).

En las primeras etapas de diseño se estudió la posibilidad de controlar el hardware del robot con la *HandyBoard* del Media Laboratory del MIT (el.www.media.mit.edu/projects/handy_board), tarjeta microcontroladora de bajo consumo basada en el *M68HC11*, programable con Interactive C (versión multitarea limitada de C) y capaz de controlar hasta cuatro motores de continua, realizar medidas digitales y analógicas, mostrar la información deseada en su pantalla LCD, y comunicarse por el puerto serie con un PC. La creciente popularidad de los microcontroladores *PIC* durante la fase del diseño del robot, debida principalmente a su versatilidad, potencia, bajo consumo, bajo

coste, y facilidad de programación, terminó por desplazar la opción de la *HandyBoard*, porque desde nuestro punto de vista ésta limitaba las opciones de su *M68HC11* al utilizar gran parte de sus puertos de entrada y salida para el soporte de la programación con Interactive C y otras opciones propias de la tarjeta.

La existencia de una amplia familia de microcontroladores *PIC*, capaces de controlar distintos tipos de dispositivos y de comunicarse entre sí y con otros dispositivos a través de distintos tipos de puertos y/o protocolos, y la construcción de la arquitectura hardware del robot de bajo nivel sobre estos microcontroladores, ha facilitado el diseño y construcción incremental de nuestro robot. Los *PIC* utilizados nos permiten controlar la velocidad de los motores en continua mediante pulsos modulados en anchura (PWM), contar los pulsos generados por unos encoders ópticos acoplados a las ruedas motrices, medir distancias utilizando el tiempo de vuelo de señales de ultrasonidos, conocer la orientación del robot utilizando una brújula que devuelve señales a través de un bus I²C, etc.

Debido al limitado repertorio de instrucciones de los *PIC* utilizados, todo el control de alto nivel del robot se realiza mediante un PC que, comunicándose con los *PIC* por el puerto serie mediante el protocolo RS-232, es el encargado de analizar toda la información sensorial recibida y de generar las señales de control necesarias. El protocolo de comunicación serie RS-232 es soportado por un amplio conjunto de ordenadores y sistemas operativos, por lo que el robot diseñado puede ser controlado por cualquier ordenador que soporte dicho protocolo.

La solución adoptada para la adquisición y procesamiento de información visual ha sido utilizar dos cámaras blanco y negro analógicas *XC-ST30CE* de Sony (www.pro.sony-europe.com/isp) junto con una tarjeta de adquisición de imágenes *Matrox* controlada desde un PC por el software de procesamiento de imágenes *MIL 6.1*. Esta solución permite desarrollar programas de procesamiento de imágenes propios; utilizar distintas cámaras con la misma tarjeta de adquisición; configurar el PC de visión con las prestaciones hardware deseadas y utilizarlo para otras labores, como el control de alto nivel, cuando se considere oportuno. El principal inconveniente de esta elección viene determinado por el software básico de tratamiento elegido, ya que la versión de las MIL (Matrox Imaging Library) utilizada es soportada por los sistemas operativos Windows (98, NT y 2000), y por lo tanto el PC de visión necesitará utilizar alguno de estos sistemas operativos.

En los apartados de este apéndice se presentan las características de los principales dispositivos del robot Nandana, comenzando con una descripción general de los diferentes elementos presentes en el robot y continuando con una descripción detallada de la arquitectura de control distribuido desarrollada, de su sistema motriz y de sus diferentes módulos sensoriales. Por último se presenta el repertorio de instrucciones desarrollado para controlar el robot desde un PC a través de su puerto serie.

A3.1. Descripción del Robot Móvil Autónomo Nandana

En este apartado se hace una descripción detallada del robot Nandana, representado de forma esquemática en la siguiente figura. Éste ha sido construido sobre una plataforma móvil *Round Max 96* de *Zagros Robotics*, que está formada por dos plataformas circulares de 40 cm de diámetro, separadas por cuatro barras circulares verticales de 17 cm, y por cuatro ruedas independientes, dos motrices de 15 cm de diámetro situadas a ambos lados del eje central del robot, y dos de castor de 7.5 cm de diámetro colocadas a ambos lados del eje ortogonal. Cada rueda motriz se encuentra acoplada a través de una caja reductora a un motor de corriente continua RS-385SH-2270 de Mabuchi Motor (www.mabuchi-motor.co.jp). Según las especificaciones de *Zagros Robotics* la velocidad máxima del robot es de 0.3 m/s cuando es alimentado con una tensión de 12 V.

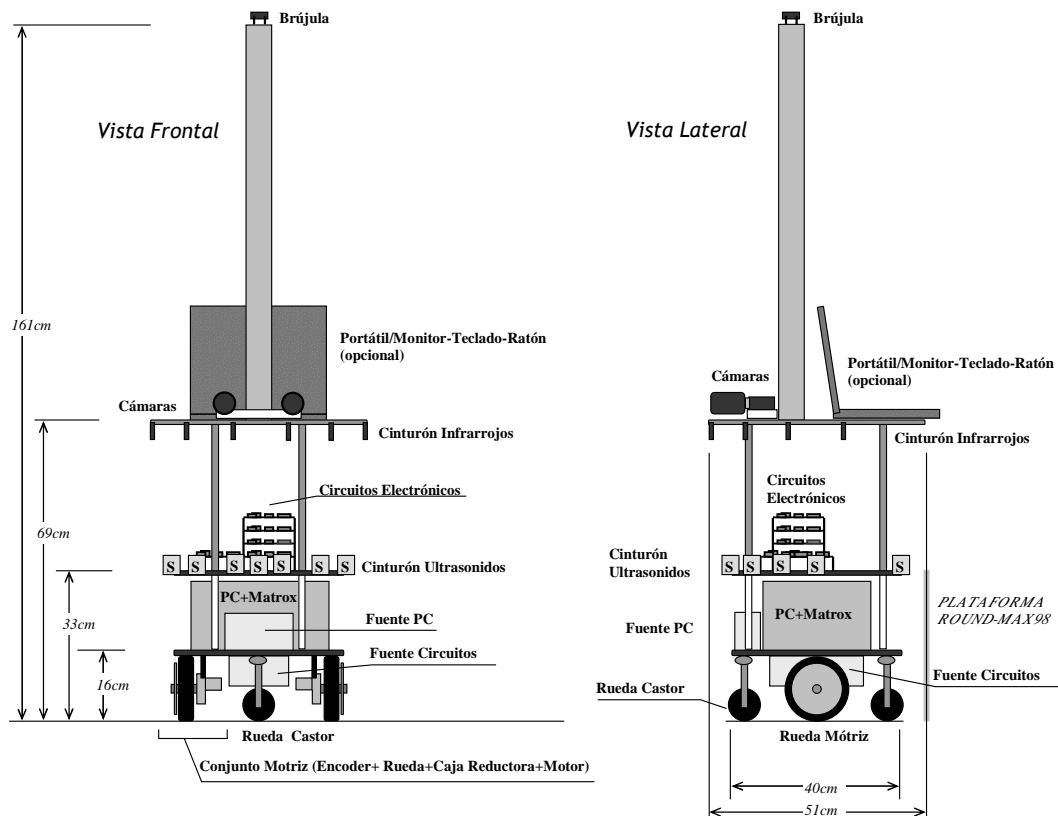


Figura A3-1. Estructura del robot móvil Nandana.

Con el fin de completar el armazón del robot se han colocado sobre la plataforma superior del *Round Max 96* una plataforma octogonal de 51 cm de diámetro, soportada por cuatro barras verticales de 36 cm, y sobre la plataforma octogonal un cuerpo cilíndrico de 92 cm de altura y 6 cm de diámetro. De esta forma, aunque la altura total sobre el suelo del robot es de 161 cm, el esqueleto básico del mismo está formado por una plataforma octogonal de 51 cm de diámetro y una altura de 69 cm, y dos plataformas circulares de 40 cm de diámetro situadas respectivamente a una

distancia de 33 y 16 cm del suelo. Sobre este armazón de tres plataformas, que divide al robot en cuatro secciones, se ha distribuido todo el hardware del robot según el esquema de la figura A3-1.

Entre el suelo y la primera plataforma circular del robot se encuentran situados el sistema motriz del *Round Max 96* (formado por las dos ruedas de castor, las dos ruedas motrices, las cajas reductoras y los motores de continua) y parte del sistema de alimentación del robot. El sistema motriz original ha sido modificado con la incorporación de un encoder incremental sobre cada rueda motriz, construido con un disco circular metálico acoplado al eje de la rueda y cuyo perímetro se encuentra perforado con 100 agujeros distribuidos uniformemente, y con un sensor optoelectrónico de infrarrojos TCST2300 (www.vishay.com) que, acoplado a la plataforma circular inferior del robot, es el responsable de detectar el paso de los agujeros mientras las ruedas giran. Con el fin de aumentar la estabilidad del robot se han colocado dos de las fuentes de alimentación independientes en el espacio libre existente entre las cuatro ruedas, una de 12 V y 2 A responsable de la alimentación de los motores, y otra que proporciona los ± 12 V y 1 A necesarios para el funcionamiento del resto de la electrónica del robot.

En el siguiente nivel, entre las dos plataformas circulares del robot, se han situado un PC fijo y su fuente de alimentación. El PC está equipado con una placa Pentium III a 866 MHz, una RAM de 256MB, una tarjeta Ethernet 10/100 Mbps, los puertos habituales de entrada y salida (2 serie, 1 paralelo, 2 USB, 2 PS2), una tarjeta gráfica *Matrox Millenium G200*, y una tarjeta de adquisición de imágenes *Matrox MeteorII/MultiChannel*. Este PC fijo puede ser utilizado para la adquisición y procesamiento de las imágenes provenientes de las cámaras (mediante el uso de la tarjeta de adquisición de imágenes y el software de procesamiento *MIL*) y/o como el PC responsable del control de alto nivel del resto de los dispositivos hardware del robot (a través de uno de sus puertos serie). Aunque desde el PC fijo se pueden realizar ambas labores simultáneamente, en los casos en los que el procesamiento de imágenes necesite hacer un uso intensivo de la CPU puede ser conveniente utilizar el PC fijo solamente para obtener los resultados de visión y utilizar un PC portátil como elemento de control de todo el sistema. En ese caso, se conectarán ambos PCs a través de un cable cruzado de red de forma que el PC portátil pueda acceder a la información extraída del proceso de visión y se utilizará el puerto serie del PC portátil para controlar el resto de los dispositivos hardware del robot.

A continuación, entre la plataforma circular superior y la plataforma octogonal, se encuentran situados la mayoría de los circuitos necesarios para el funcionamiento del robot, un cinturón de sensores de ultrasonidos (situado en la plataforma circular) y otro cinturón de sensores de infrarrojos (situado en la plataforma octogonal). El cinturón de infrarrojos se encuentra formado por 8 sensores *GP2D12* de *Sharp* (www.sharpsma.com), cuyo funcionamiento se basa en la medida de la posición iluminada sobre el elemento receptor, y que pueden ser empleados para medir la distancia a objetos situados entre 10 y 80 cm. El cinturón de ultrasonidos está formado por 8 pares

de emisores-receptores de ultrasonidos *MA40S4R/S* de *Murata* (www.murata.com) junto con sus correspondientes circuitos de excitación, filtrado y detección, contruidos de acuerdo a las especificaciones de los sensores del cinturón de ultrasonidos del robot Mar IV de la Universidad de Castilla la Mancha. Éstos son utilizados para calcular la distancia a objetos en función del tiempo existente entre la emisión y recepción de un pulso de ultrasonidos, siendo capaces de detectar objetos situados entre 20 centímetros y unos 2-3.5 metros.

La electrónica necesaria para controlar los dispositivos hardware del robot (salvo las cámaras de visión), situada en este nivel, ha sido desarrollada en torno a un grupo de *PIC16F873/6*, microcontrolador caracterizado por su versatilidad, su fácil uso y programación, y su bajo consumo y precio. Entre sus múltiples interfaces de comunicación con el exterior, su módulo USART de transmisión direccionable permite construir un protocolo de comunicación serie asíncrona entre un PC de control y un conjunto de *PIC16F873/6*, identificados cada uno con una dirección, y por lo tanto construir una arquitectura hardware de control distribuido de bajo nivel, donde cada microcontrolador es responsable del control de una parte del hardware del robot. Siguiendo esta filosofía, es sencillo diseñar un robot abierto a la inclusión de nuevos dispositivos (controlados desde nuevos *PIC16F873/6*), y adaptar a nuevas exigencias la funcionalidad de los módulos existentes (modificando el programa del *PIC16F873/6* correspondiente). De acuerdo con éste control distribuido, la electrónica del robot Nandana, se encuentran dividida en electrónica responsable de: controlar el movimiento del motor (asociada al *PICMotores*), medir distancias utilizando el cinturón ultrasonidos (asociada al *PICSonar*), y medir la orientación del robot y distancias utilizando el cinturón de infrarrojos (asociada al *PIC²C* ya que ambos tipos de medidas son recogidas por dispositivos que utilizan el bus I²C para intercambiar información). Como ya se ha mencionado anteriormente, el hardware del robot puede ser controlado tanto desde el PC fijo situado sobre la plataforma circular inferior como desde un PC portátil, siendo únicamente necesario conectar el terminal del PC de la conexión serie PC-conjunto de PIC al PC elegido, y utilizar desde dicho PC el protocolo de comunicación desarrollado para el control del hardware del robot.

En el nivel superior, encima de la plataforma octogonal, se han situado dos cámaras blanco y negro analógicas *XC-ST30CE* de *Sony* y una columna cilíndrica de 92 cm de altura sobre la que está colocada una brújula electrónica *CMPS03* de *Devantech Ltd* (www.robot-electronics.co.uk). Las cámaras se encuentran alineadas sobre una pequeña plataforma rectangular y conectadas a la tarjeta de adquisición *Matrox MeteorII/MultiChannel* de forma que se pueden capturar simultáneamente las imágenes procedentes de ambas cámaras, configuración elegida porque permite extraer información visual estereoscópica (a partir de un par de imágenes capturadas por cada cámara) y de movimiento (a partir de series de imágenes tomadas por cada cámara). La brújula se ha colocado en

lo alto de la columna para minimizar las variaciones del campo magnético causadas por el resto de los dispositivos electrónicos del robot.

Por último, sobre la plataforma octogonal se puede colocar un PC portátil o un monitor plano, teclado y ratón para el control del PC fijo, y/u otros dispositivos sensoriales que se deseen utilizar en el robot, como una torreta giratoria de ultrasonidos que, construida para este robot con una lógica similar a la del cinturón de ultrasonidos y controlable por el microcontrolador *PIC Sistema Triular*, permite orientar un sistema triular de ultrasonidos para obtener la distancia y orientación de objetos próximos. En la figura A3-2 se presentan dos fotografías del robot Nandana, correspondientes a la vista frontal y lateral recogidas en la figura A3-1, con un portátil sobre la plataforma octogonal superior.

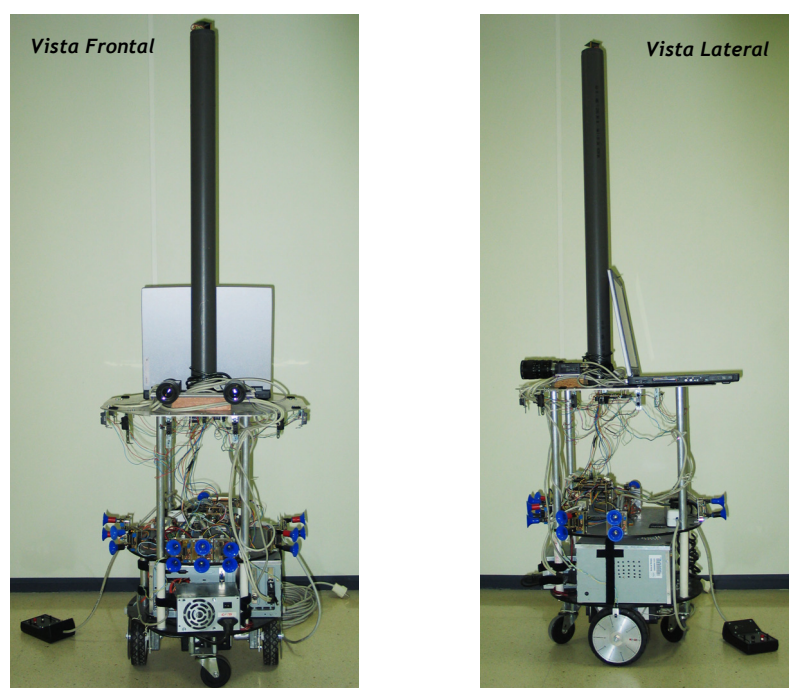


Figura A3-2. El robot móvil Nandana.

A3.2. Arquitectura de Control

El control del robot se encuentra distribuido entre un conjunto de microcontroladores *PIC16F873/6* (responsables de realizar el control de bajo nivel de los dispositivos hardware del robot, recoger la información procedentes de los distintos sensores y enviar las ordenes a los actuadores), un PC fijo equipado con una tarjeta de adquisición de imágenes *Matrox MeteorII/MultiChannel* (encargado de la captura y procesamiento de imágenes y/o del control de alto nivel del robot), y opcionalmente de un PC portátil (configurado como elemento de control central). El PC de control (fijo o portátil) y los microcontroladores se comunican a través de un cable serie haciendo uso del puerto serie del robot, la USART direccionable de los *PIC16F873/6* y el protocolo de comunicación RS-232.

Además, cuando los dos PC están funcionando, el intercambio de información entre los mismos se realiza a través de un cable cruzado de red conectado a sus respectivas tarjetas Ethernet. Por último, dentro de la arquitectura hardware de bajo nivel, algunos dispositivos se comunican con los *PIC16F873/6* utilizando el protocolo de comunicación serie I²C.

La posibilidad de utilizar dos PC como elemento principal de control permite múltiples configuraciones, dos de las cuales, en las que no aparece la lógica asociada a la torreta de ultrasonidos, se recogen en los diagramas de bloques de la figura A3-3.

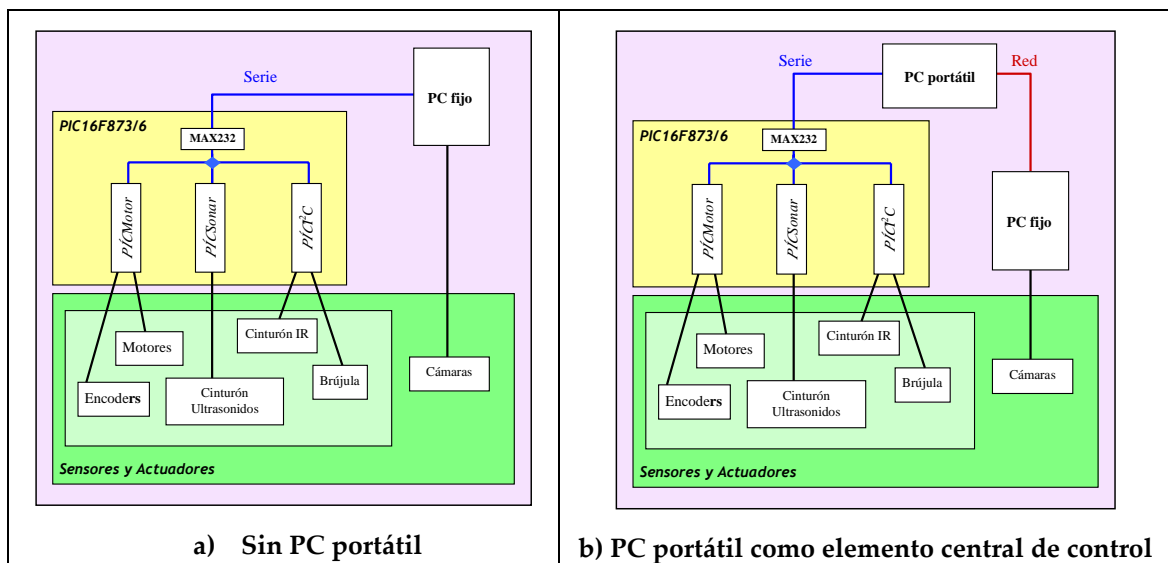


Figura A3-3. Diagramas de bloques de posibles arquitecturas de control.

En todas las arquitecturas de control las cámaras se encuentran conectadas al PC fijo mediante la tarjeta *Matrox MeteorII/MultiChannel* y el resto del hardware del robot al grupo de *PIC16F873/6* responsables del control distribuido de bajo nivel. La diferencia entre las mismas radica en el PC elegido para conectar los microcontroladores:

- La configuración sin PC portátil, recogida en figura A3-3.a), permite controlar completamente el robot desde el PC fijo. Aunque evita el intercambio de información entre los dos PC, el control en tiempo real puede verse degradado en los casos en los que sea necesario hacer un uso intensivo de la CPU, tanto para el procesamiento de imágenes, como para el procesamiento del resto de la información sensorial y la toma de decisiones.
- En la configuración con PC portátil como elemento central de control, representada en A3-3.b), dicho PC es el responsable de controlar el robot a partir de la información visual preprocesada por el PC fijo de las cámaras y la información sensorial del resto de los dispositivos hardware conectados a los *PIC16F873/6*. En este caso, el conjunto PC fijo y cámaras funciona como un sensor de alto nivel, que en vez de mandarle al PC portátil las

imágenes directamente, le envía la información visual extraída por sus algoritmos de procesamiento, liberando a la CPU del PC portátil de dicha labor.

A3.2.1. Arquitectura Hardware Distribuida de Bajo Nivel basada en el PIC16F873/6.

Los microcontroladores PIC16F873/6 sobre los que se ha construido la arquitectura hardware de bajo nivel se caracterizan por su gran versatilidad debido a sus múltiples posibilidades de funcionamiento, bajo consumo, rapidez, bajo coste y facilidad de programación (en nuestro caso, en ensamblador, haciendo uso del compilador MPLab y el software de descarga ICProg). Las principales especificaciones de este microcontrolador, cuyo funcionamiento se encuentra detallado en [Pic16F87x] y [PicMR], se resumen en las dos tablas siguientes.

En la primera (tabla A3-1) se presentan las características de su núcleo, construido sobre una CPU RISC de altas prestaciones, cuya velocidad máxima de 20 MHz puede ser definida por medio de un oscilador externo, y sobre sus tres bancos de memoria (programa, datos y EEPROM) de alta velocidad y bajo consumo.

Programación	CPU RISC (35 instrucciones) de altas prestaciones
	Ejecución de 1 instrucción por ciclo, salvo las de salto (2 ciclos)
	14 fuentes de interrupción
	Pila hardware de 8 niveles
	Direccionamiento: directo, indirecto y relativo
	Reset y Timer de encendido
	Temporizador WatchDog con oscilador RC incorporado en el PIC
	Posibilidad de escribir en la memoria del programa durante la ejecución
	Programación serie In-Circuit (ICSP) a través de 2 Pines/single 5 V In-Circuit serie
	Depuración In-Circuit a través de 2 pines
	Protección de código opcional
Velocidad	Alta velocidad: hasta 20MHz, con 1 ciclo de instrucción cada 4 pulsos (200ns)
	Opciones del oscilador responsable de la velocidad configurables
Memoria	Programa almacenado en memoria Flash (hasta 8K palabras de 14 bits)
	RAM de datos: hasta 368 palabras de 1 byte
	EEPROM: hasta 256 palabras de 1 byte
Alimentación y Consumo	Alimentación: 2.0 a 5.0 V
	Gran capacidad de aporte/consumo de corriente (25mA)
	Consumo: < 0.6mA típicamente a 3 V, 4 MHz < 20µA típicamente a 3 V, 32 KHz < 1µA típicamente en modo ahorro de energía
	Modo ahorro de energía
Otros	Tecnología CMOS Flash/EEProm de alta velocidad y bajo consumo
	Rango de temperaturas de trabajo industrial/comercial

Tabla A3-1. Características del núcleo del PIC16F873/6

En la segunda (tabla A3-2) se muestra el conjunto de módulos adicionales de este microcontrolador junto con sus puertos de entrada/salida multipropósito configurables y los protocolos asociados a

los mismos, y se indica cuales han sido usados en los *PIC16F873/6* que controlan el hardware básico del robot: el *PICMotores* (columna M), responsable del control del movimiento del robot; el *PICSonar* (columna S), utilizado para medir distancias por medio del cinturón de ultrasonidos; y el *PIC²C* (columna I), usado para comunicarse con la brújula y controlar el hardware asociado al cinturón de infrarrojos.

Módulos y puertos de E/S			PICS		
			M	S	I
Timer	Timer0 (8 bits, preescaler)	Temporizador			
		Contador	✓		
	Timer1 (16 bits, preescaler)	Temporizador		✓	
		Contador	✓		
	Timer2 (8 bits, preescaler y postscaler)	Temporizador	✓	✓	✓
		Contador			
2 pines configurables en modo	Capturador: almacena el valor del Timer1 ante un evento en el pin			✓	
	Comparación: si el Timer1 llega a un valor hay un evento en el pin				
	PWM: pulso modulado en anchura en el pin (utilizando el Timer2)		✓		
Puerto Serie Sincrono (SSP)	SPI (Serial Peripheral Interface)				
	I ² C (Inter. Integrated Circuit)				✓
Transmisor/Receptor Sincrono/Asincrono Universal (USART)	USART en modo asincrono	Normal	✓	✓	✓
		Direccionable	✓	✓	✓
	USART en modo master sincrono				
	USART en modo esclavo sincrono				
Entradas/ Salidas	Entadas/Salidas Digitales: Hasta 22 (divididos en 3 puertos)		✓	✓	✓
	Entradas Analógicas: hasta 6, (convertor A/D de 10 bits)		✓		

Tabla A3-2. Módulos adicionales y puertos de E/S del *PIC16F873/6*

A3.2.2. Comunicación entre el PC y los Microcontroladores: Protocolo Serie con Auto Identificación del Receptor

El protocolo de comunicación entre el PC y el conjunto de *PIC16F873/6*, esquematizado en los diagramas de bloques de la figura A3-4, se fundamenta en las siguientes características de los módulos USART de ambos dispositivos:

- Entre las múltiples configuraciones posibles de la USART del *PIC16F873/6* se encuentran la transmisión asíncrona de palabras de 9 bits, con detección de dirección en la recepción (USART asíncrona direccionable) o sin ella (USART asíncrona normal). En el primer caso, la USART hace uso del valor bit9 para determinar si la palabra recibida debe de ser desechada (bit9=0) o almacenada (bit9=1) en el registro de recepción que el *PIC16F873/6* utiliza para acceder a la información recibida, mientras que en el segundo caso la USART no comprueba el valor del bit9 y vuelca en el registro de recepción todas las palabras recibidas.

En ambas configuraciones se puede habilitar la interrupción asociada al volcado de una palabra en el registro de recepción.

- La USART del PC soporta la transmisión asíncrona de palabras de 9 bits, 8 de información y 1 de paridad, cuyo valor es calculado por el módulo USART de acuerdo con la paridad elegida a la hora de habilitar el módulo y los restantes 8 bits de datos. Es posible elegir el valor del bit de paridad de cada palabra enviada utilizando únicamente 7 de los 8 bits de datos para el intercambio de información, y el octavo para forzar que el bit9 tenga el valor deseado. Siguiendo este método para seleccionar el valor del bit de paridad, las palabras enviadas desde el PC a los *PIC16F873/6* tienen únicamente 7 bits de información útil, mientras que las palabras enviadas desde los microcontroladores al PC pueden hacer uso de los 8 bits habituales.

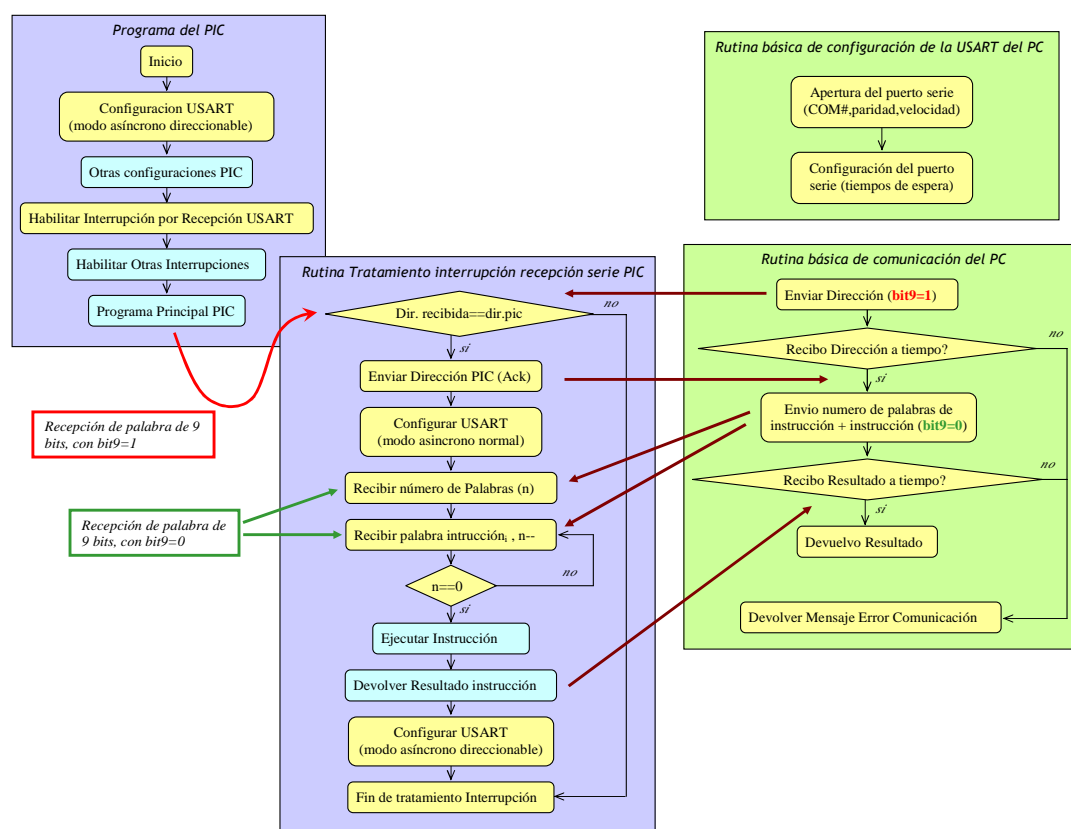


Figura A3-4. Diagramas de bloques de las rutinas del protocolo de comunicación PC-PIC16F873/6.

El protocolo de comunicación implementado para la comunicación entre el PC y el conjunto de *PIC16F873/6* es dirigido desde el PC, que antes de mandar una instrucción envía una palabra de 9 bits, cuyo bit9 vale 1, para identificar al microcontrolador con el que se quiere comunicar. Todos los microcontroladores, cuya USART está configurada en modo asíncrono direccionable, reciben la palabra enviada lo que provoca un salto a la rutina de tratamiento de interrupción correspondiente. En dicha rutina, se compara la dirección enviada desde el PC con la dirección almacenada en el

PIC16F873/6 y en caso de ser la misma, el microcontrolador reconfigura su USART en modo asíncrono normal y envía la dirección al PC, para indicarle que uno de los microcontroladores ha sido identificado y está preparado para recibir el resto de la instrucción. Una vez que el PC recibe la dirección enviada por el *PIC16F873/6*, envía toda la información relativa a la instrucción dividida en varias palabras de 9 bits, fijando el valor del bit9 a 0, de forma que únicamente el microcontrolador identificado es capaz de recibir dicha información. El resto de los microcontroladores tienen la USART configurada en modo asíncrono direccionable, por lo que la información enviada por el PC no es volcada en el registro de recepción y no se produce un salto a la rutina de tratamiento de interrupción. Una vez que el microcontrolador elegido (direccionado) recibe toda la información relativa a la instrucción, ésta es ejecutada en el microcontrolador. El resultado de la misma es enviada al PC, y la USART del microcontrolador se reconfigura en modo asíncrono direccionable.

La información relativa a una instrucción se envía a cada uno de los microcontroladores utilizando siempre el mismo formato: la primera palabra indica el número de palabras que constituyen la instrucción, la siguiente palabra identifica la instrucción enviada, y el resto incluye cualquier información adicional relativa a la misma. Según el microcontrolador y la instrucción enviada, se ejecuta una rutina de tratamiento de instrucción diferente y se devuelve el resultado de la misma en tantos bytes como sea necesario. En el último apartado de este apéndice, tras explicar el funcionamiento de los sistemas sensoriales y de control del robot, se presenta el repertorio de instrucciones asociado a cada uno de los microcontroladores del robot.

Las características del protocolo de comunicación implementado permiten a un PC comunicarse con un número elevado de microcontroladores, y por lo tanto realizar un control de bajo nivel del hardware del robot distribuido entre tantos *PIC16F873/6* como sea necesario, e incorporar nuevos dispositivos controlables desde nuevos microcontroladores siempre que se desee.

Por último, es importante poner de manifiesto que debido a los diferentes niveles de señal existentes entre la USART del PC (niveles RS-232) y la USART de los microcontroladores (niveles TTL) es necesario añadir una etapa intermedia responsable de la conversión de niveles, motivo por el que se ha colocado en la electrónica del robot un *MAX232* de *MAXIM* (www.maxim-ic.com) entre las líneas serie RX y TX del PC y los *PIC16F873/6* (incluido en el esquema de figura A3-3). Además, las líneas de transmisión de los microcontroladores no pueden ser conectadas directamente a un único punto a la entrada del *MAX232*, si no que es necesario incluir una puerta triestado (*74HC126*) entre cada una de las líneas y el punto común de conexión. Dicha puerta es controlada con el microcontrolador a través de uno de los pines de sus puertos, asegurando que la salida de la puerta está en alta impedancia mientras el microcontrolador no es direccionado y tiene que comunicarse con el PC.

A3.2.3. Comunicación entre el PC Fijo y el PC Portátil: Sockets TCP/IP

El intercambio de información entre el PC fijo y el PC portátil se realiza a través de un cable de red cruzado conectado a sus respectivas tarjetas de Ethernet. Entre los múltiples protocolos disponibles se han utilizado los sockets TCP/IP, ya que es un método de comunicación estándar entre PCs que se encuentra implementado para la mayoría de los sistemas operativos existentes.

A3.2.4. Comunicación entre los microcontroladores y Dispositivos I2C: Protocolo de Comunicación Serie I2C

El protocolo serie I²C [I2C_Specs], utilizado por algunos dispositivos electrónicos para intercambiar información con los microcontroladores, permite conectar un conjunto de varios dispositivos (uno actuando como maestro y el resto como esclavos identificados con direcciones diferentes) al mismo bus (formado por una línea SDA de datos y otra SCA de reloj, ambas con sus correspondientes resistencias de pull-up). Cada vez que el maestro inicia una comunicación escribe una palabra de 8 bits, utilizando los 7 más significativos para direccionar el dispositivo elegido, y el restante para indicar si quiere enviar un dato al dispositivo (escritura) o recibir información del mismo (lectura). A continuación, en el caso de una escritura se escribe la orden como una palabra de 8 bits, y en el caso de una lectura se leen tantas palabras de 8 bits como se considere oportuno. Cuando se desea que un dispositivo devuelva información, primero se le indica por una orden de escritura la información que se le solicita, y posteriormente mediante una orden de lectura se lee la información repartida en tantas palabras como sea necesario. En la figura A3-5 se presenta esquemáticamente el funcionamiento de este protocolo para un dispositivo de dirección en escritura es 0xC0 y en lectura 0xC1.

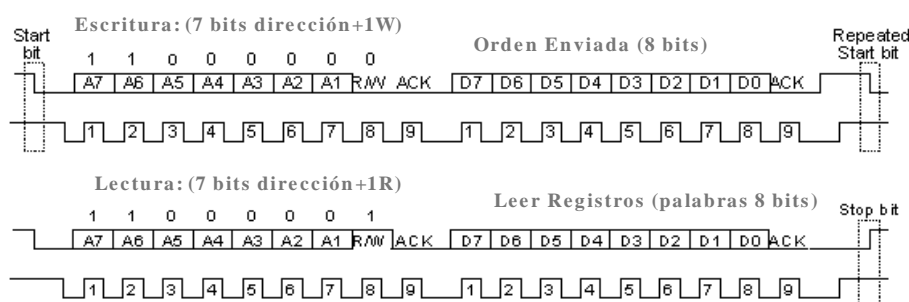


Figura A3-5. Protocolo de comunicación serie I²C.

A3.3. Sistema Motriz

El sistema motriz del robot Nandana está constituido por dos ruedas independientes de 15 cm de diámetro, acopladas por un lado a dos motores de corriente continua mediante sus correspondientes

cajas reductoras, y por el otro a dos encoders incrementales formados por dos discos metálicos perforados uniformemente y dos sensores optoeléctricos.

Los motores del robot soportan tensiones de alimentación de 6 a 24 V, y alimentados a 12 V tienen un consumo medio de 0.75 A en la zona de máxima eficiencia y un pico de 3.9 A en el arranque. Para evitar el aumento de consumo a tensiones superiores, los 12 V son la tensión máxima de alimentación elegida para los motores del robot. A dicha tensión, una vez conectados a cada rueda por medio de su correspondiente caja reductora, tienen un consumo de 1.3 A, son capaces de proporcionar un torqué de 2.225 Nm y mover al robot con una velocidad máxima de 0.2 m/s.

El control de la velocidad de giro de las ruedas motrices es realizado por el microcontrolador *PICMotores* que genera, para cada motor y de forma totalmente independiente, dos señales TTL: (1) el sentido de giro del motor y (2) un pulso modulado en anchura (señal PWM) responsable de determinar la velocidad del robot. Ambas señales son calculadas por el microcontrolador en función de las velocidades solicitadas desde el PC de control a través del puerto serie (control automático) o de las tensiones de referencia proporcionadas por los dos potenciómetros lineales del joystick del robot (control manual), y se encuentran conectadas a una etapa de potencia responsable de suministrar la tensión y la corriente necesaria para alimentar al motor. Un circuito de encendido y apagado permite inhibir el funcionamiento de los motores en caso de emergencia por medio de dos botones del joystick. El diagrama de bloques del cuerpo motriz del robot se representa en la figura A3-6.

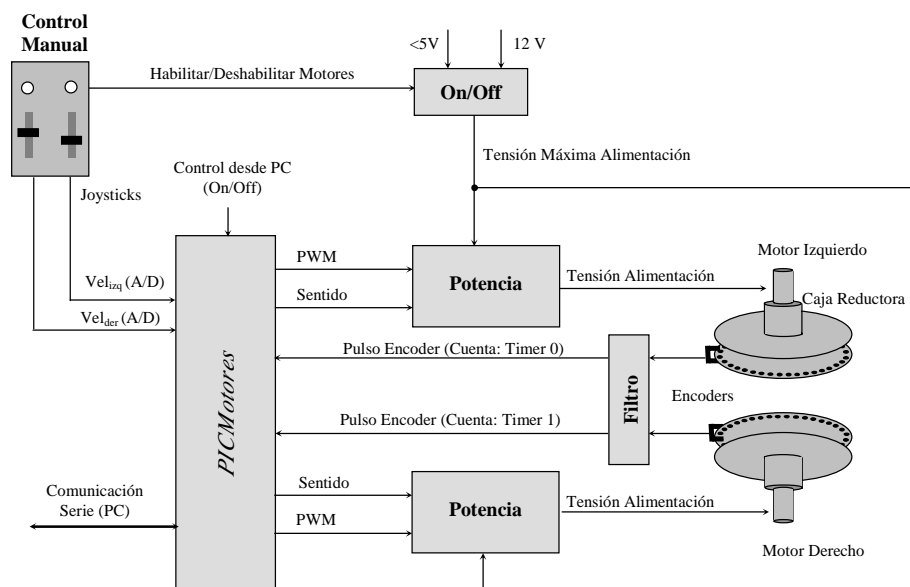


Figura A3-6. Diagrama lógico de control del sistema motriz y de los encoders.

La etapa de potencia, construida sobre un circuito integrado L293B (capaz de proporcionar una intensidad de 1 A, con picos de hasta 2 A), alimenta a cada motor con una tensión promedio

responsable de la velocidad de las ruedas, proporcional a la tensión máxima de alimentación conectada a la etapa y al porcentaje de tiempo en el que el pulso modulado en anchura tiene un valor lógico de 1. El signo de la diferencia de tensión entre los bornes del motor, responsable del sentido de giro de las ruedas, viene determinado por la señal de sentido proporcionada por el microcontrolador.

La etapa de potencia se inhibe cuando la tensión de alimentación de la lógica del L293B (5 V) es superior a la tensión máxima de alimentación, comportamiento que se utiliza para deshabilitar los motores en caso de emergencia. El circuito de encendido y apagado, construido sobre un relé y controlado por los dos botones del joystick, es el responsable de determinar la tensión máxima de alimentación conectada a la etapa de tensión: tras pulsar el botón de encendido la tensión es de 12 V (valor elegido como tensión máxima de alimentación) y después de apretar el botón de apagado la tensión es inferior a 3 V (lo que deshabilita la etapa de potencia provocando la parada inmediata de los motores).

Los pulsos modulados en anchura son generados por los dos módulos de PWM del microcontrolador, configurados de forma que la anchura del pulso sea definida a través de palabras de 7 bits, lo que permite generar hasta 128 señales PWM con diferente ancho de pulso y, por lo tanto, solicitar a las ruedas hasta un máximo de 128 velocidades distintas (256 en el caso de considerar también el sentido de giro). La anchura del pulso es elegida basándose en la tensión de referencia del joystick (control manual) o en las órdenes de velocidad enviadas desde el PC central (control automático).

En el caso del control manual las tensiones de referencia procedentes de los potenciómetros lineales del joystick toman valores en el intervalo de 0 a 5 V y son capturadas por el conversor analógico digital del *PICMotores*, que tiene una precisión de 10 bits. Para controlar tanto la velocidad como el sentido de giro, se han utilizado únicamente los 8 bits más significativos de la conversión, de forma que en caso de que el bit más significativo sea 1 (tensiones por encima de 2.5 V) la rueda gire en sentido positivo a la velocidad indicada por los restantes 7 bits, y en el caso contrario (tensiones inferiores a 2.5 V), la rueda gire en sentido negativo a la velocidad indicada por el complemento a uno de los 7 bits restantes. De esta forma, el giro nulo del motor se realiza a los 2.5 V, la velocidad máxima para adelante a los 5 V y la velocidad máxima hacia atrás a los 0 V.

En el caso de realizarse un control automático de velocidades, el PC utilizará tres palabras para las instrucciones de velocidades: en la primera, responsable de que el microcontrolador la identifique como una orden de cambio de velocidad se incluirá también el sentido de las dos ruedas (codificados en los dos bits menos significativos); y en las dos siguientes, la velocidad de cada una (codificados en los siete bits útiles de cada palabra de datos).

De los 128 pulsos de la señal de PWM, hay un conjunto que no es capaz de suministrar la tensión promedio suficiente para arrancar los motores, por lo que no son utilizados en el control automático. Además, los motores de continua son sistemas no lineales, lo que obliga a hacer un estudio para caracterizar la respuesta en velocidad del sistema ante diferentes pulsos de entrada. En la figura A3-7 se recogen los resultados obtenidos, en una gráfica que presenta, para cada rueda y sentido de giro, la velocidad de giro en función del pulso elegido. Además, en la misma figura se representa la función elegida para modelar la velocidad de respuesta de las ruedas ante el valor de la señal de PWM aplicada.

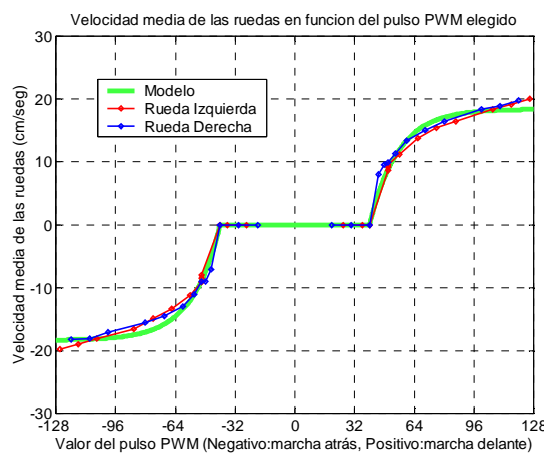


Figura A3-7. Velocidad de los motores en función de la señal PWM elegida.

Por último es importante reseñar que la elección del tipo de control se realiza por medio de un interruptor conectado a uno de los pines de entrada del microcontrolador. En una de sus posiciones habilita directamente el control manual, para poder desplazar el robot sin encender el PC de control. En la otra, se habilita la comunicación serie entre el PC y el microcontrolador, y por medio de instrucciones enviadas desde el PC, se habilitan/deshabilitan los motores, y se decide si el control que se va a utilizar es manual/automático.

A3.4. Sensorización

Con el objetivo de poder estudiar el funcionamiento de los algoritmos de fusión multisensorial implementados durante el desarrollo de esta tesis, se ha dotado al robot de un conjunto de módulos sensoriales, redundantes y complementarios, de cuyas medidas se puede extraer información relativa a la posición del robot y a la posición de los objetos en su entorno.

En este apartado se describe en detalle el funcionamiento y características de los módulos ya incorporados: los encoders incrementales acoplados a las ruedas mótices, la brújula electrónica, el cinturón de infrarojos, el cinturón de ultrasonidos y el sistema de visión. Por último se presenta

un sistema adicional triular orientable de ultrasonidos, que se puede colocar encima del robot y controlar desde la arquitectura hardware de control.

A3.4.1. Encoders Ópticos Incrementales

Cada una de las ruedas motrices del robot lleva acoplada un disco circular metálico de 12 cm de diámetro, perforado con 100 agujeros (de 1.5 mm de diámetro) distribuidos uniformemente, cada 3.6°, a lo largo de una circunferencia próxima a su perímetro (de 11 cm de diámetro). Se utiliza un sensor optoelectrónico de infrarojos TCST2300, cuyo emisor y receptor se encuentran situados frente a frente a lo largo del mismo eje óptico y que tiene una resolución de 0.2 mm. Este sensor se ha colocado debajo de la plataforma inferior del robot de forma que el disco y sus agujeros se encuentran entre su emisor y receptor. De esta manera se puede obtener una señal de tensión que permite determinar si entre el emisor y el receptor hay un agujero del disco o no, y cuando se pasa de un estado al otro. La disposición de los componentes descritos se puede ver en la figura A3-8.

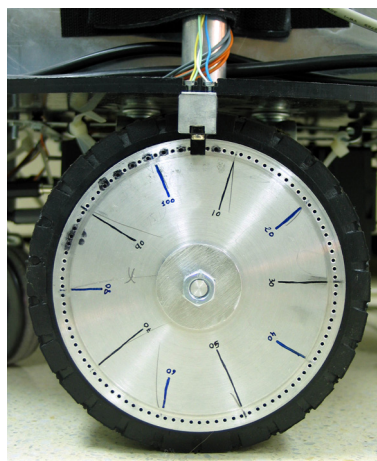


Figura A3-8. Encoder óptico incremental.

Para construir el encoder incremental es necesario utilizar únicamente algún dispositivo capaz de llevar la cuenta del número de flancos (cambios de estado) de la señal. Los timers 0 y 1 del *PICMotores* pueden ser configurados como contadores capaces de detectar flancos de subida o bajada en alguno de sus pines, por lo que han sido utilizados para llevar la cuenta de las señales generadas por el giro de ambas ruedas. El sentido de giro no es tenido en cuenta por el encoder construido, incrementándose la cuenta tanto si la rueda gira en un sentido como en el contrario.

La resolución del timer1 es de 16 bits lo que le permite contar hasta 65536 pulsos antes de que desborde, mientras que la resolución del timer0 es sólo de 8 bits y únicamente puede almacenar la cuenta de 256 pulsos. Para aumentar la resolución del timer0 y hacerla similar a la del timer1 se ha habilitado la interrupción por desbordamiento del timer0 y se utiliza un registro de 8 bits, cuyo valor se incrementa cada vez que se produce dicha interrupción. De esta forma, el encoder

incremental construido tiene una resolución de 100 pulsos por vuelta y utiliza 16 bits para almacenar la cuenta de cada una de las ruedas.

Tras construir los encoders se detectó que en algunos casos la cuenta del timer1, cuyo pin de entrada presenta una elevada sensibilidad ante pequeños cambios de nivel, se incrementaba a pesar de que el robot permaneciese parado. Tras analizar el problema se observó la aparición de un pequeño rizado en la señal de salida del sensor optoelectrónico, cuyos flancos (cambios de nivel lógico) eran interpretados por el microcontrolador como pulsos producidos por el movimiento de la rueda. Para corregir el problema se ha añadido entre el sensor y el microcontrolador *PICMotores* un filtro digital que elimina el rizado de las señales. En la figura A3-6, junto con el sistema motriz del robot, se representa el diagrama de bloques de la electrónica y elementos hardware asociados a los encoders incrementales.

Desde el PC de control, a través de un repertorio de instrucciones asociadas a este microcontrolador, se puede habilitar y deshabilitar el funcionamiento de los encoders (habilitando y deshabilitando la cuenta de los timers), resetear la cuenta de los mismos (iniciando la cuenta de los timers) y solicitar el valor de la cuenta almacenada hasta ese momento. A la hora de utilizar la cuenta de los encoders hay que recordar que ésta se incrementa para los dos sentidos de giro, y que por lo tanto no es posible para el PC determinar el sentido a partir de esta cuenta. Para permitir al PC conocer el sentido de giro se ha incluido en el repertorio de instrucciones del *PICMotores* una instrucción de lectura de encoders y sentido de giro (valores almacenados en el microcontrolador).

A3.4.2. Brújula Electrónica

Para determinar la orientación del robot en todo instante, se ha incorporado a la electrónica del robot una brújula electrónica *CMPS03* de *Devantech* que utiliza dos sensores de campo magnético *KMZ51* de *Philips* (www.semiconductors.philips.com) dispuestos perpendicularmente entre sí con el objetivo de detectar la componente horizontal del campo magnético de la tierra, y un microcontrolador *PIC16F872* responsable de capturar los valores proporcionados por ambos sensores y de realizar los cálculos necesarios para devolver la orientación del robot como un pulso modulado en anchura (formato PWM) o a través de un bus I²C. En nuestro caso utilizamos el segundo formato, ya que aunque ambas señales pueden ser leídas por los *PIC16F873/6*, este microcontrolador dispone de un módulo de comunicación I²C pero no de un módulo de captura de PWM por lo que sería necesario implementar una rutina para la señal PWM. En la figura A3-9 se presenta una fotografía de la brújula utilizada y en la figura A3-10 el diagrama de conexión de la misma con el microcontrolador *PIC²C*, responsable de la comunicación y control de todos los dispositivos que intercambian información por medio del bus I²C.

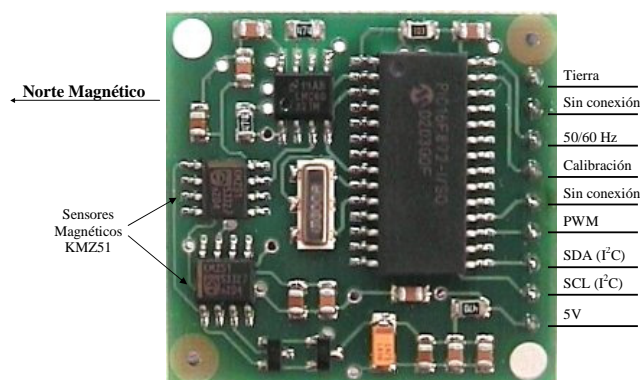


Figura A3-9. Brújula electrónica

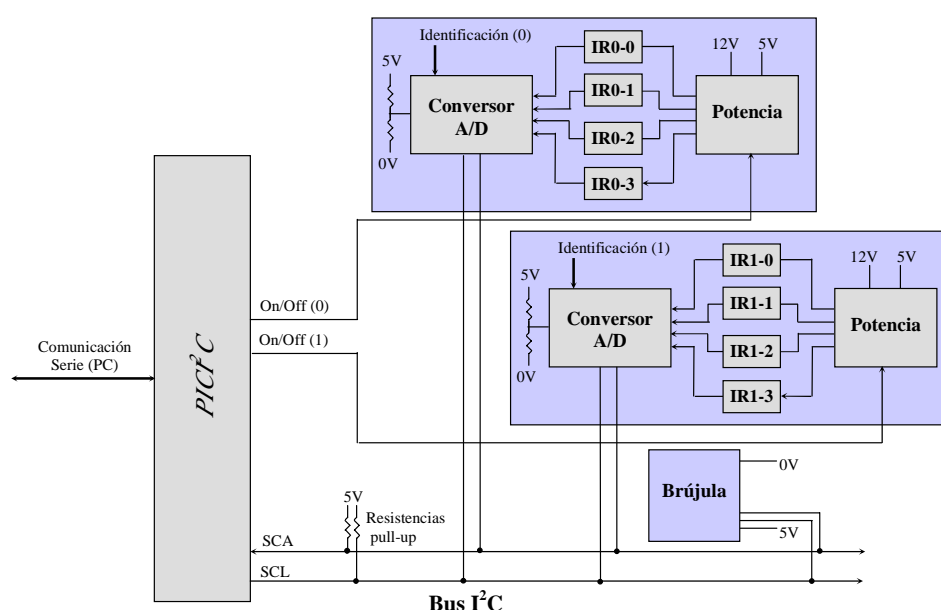


Figura A3-10. Diagrama lógico del circuito de control de la brújula y del cinturón de infrarrojos

La dirección I²C de este dispositivo es 0xC0 en escritura y 0xC1 en lectura, y la orientación medida por el sistema se encuentra almacenada en dos formatos, en un único registro de 8 bits (0-255 correspondientes a 0-359.9°) o en dos registros de 8 bits (0-3599 correspondientes a 0-359.9°). El acceso a los valores de estos registros se realiza a través de bus I²C por medio de una orden de escritura en la que se direcciona el registro y una orden de lectura en la que se lee el valor del registro direccionado. La velocidad del bus I²C para este dispositivo puede llegar hasta un 1.3 MHz aunque la comunicación con el PIC16F72 se ha fijado a la velocidad de 100 KHz.

Además de la lectura de los tres registros donde se almacena la orientación del dispositivo es posible acceder a través del bus I²C a los registros del microcontrolador propio de la brújula donde se almacenan las medidas proporcionadas por los dos sensores de campo magnético y los valores de calibración utilizados para obtener la medida de orientación. También se puede calibrar la

brújula, tomando las medida en las cuatro direcciones de orientación principales (N, S, E, W), bien a través de una orden enviada por el puerto I²C, o de uno de sus pines de conexión con el exterior.

La brújula elegida se caracteriza por tener un pequeño tamaño (34×32 mm), su bajo consumo (15 mA), utilizar una tensión alimentación de 5 V, proporcionar medidas con una resolución de 3° y una precisión máxima de 0.1° , y tener una medida disponible cada 40 o 33.3 ms (frecuencia de medida seleccionable a través de su pin de 50/60 Hz) a la que se puede acceder a través del bus I²C por medio de la lectura de los dos registros que almacenan la información de mayor precisión (valores 0-3599). Dicha lectura es solicitada desde el *PIC²C* (maestro del bus I²C) cada vez que este microcontrolador recibe, a través de su puerto serie (USART), la instrucción de petición de orientación enviada por el PC de control. Como respuesta a dicha instrucción, el PC recibirá a través del mismo puerto serie el valor de los dos registros de la brújula leídos por el microcontrolador a través del bus I²C. Por lo tanto, en este caso, el microcontrolador únicamente funciona como interfaz entre el puerto serie y el bus I²C.

Por último, la brújula se encuentra separada del resto del robot por una columna cilíndrica de 92 cm de altura, para evitar las perturbaciones que los motores, PC y demás dispositivos electrónicos del robot producen sobre el campo magnético leído por la brújula. Sin embargo, esta disposición no evita las perturbaciones producidas por los dispositivos electrónicos existentes dentro de un edificio, por lo que la navegación en interiores utilizando una brújula presenta serios inconvenientes.

A3.4.3. Cinturón de Infrarrojos

El cinturón de infrarrojos se encuentra formado por 8 sensores *GP2D12* de *Sharp*, muy utilizados en robótica debido a su bajo precio y escaso peso y tamaño ($40 \times 14 \times 13$ mm). Cada uno de los sensores está formado por un LED emisor de infrarrojos y un receptor PSD (Position Sensing Device) que es utilizado para determinar la distancia a los objetos reflectores situados entre 10 y 80 cm aplicando el principio de triangulación sobre la posición de los elementos receptores iluminados por la luz reflejada. Este sensor funciona con una alimentación (V_{CC}) de 4.5 a 5.5 V, siendo su consumo medio de 35 mA (y el pico de 200 mA), y devuelve una tensión (V_{OUT}) en función a la distancia entre el sensor y el objeto reflector cada 52 ms. A partir de esta tensión se puede calcular la distancia utilizando como función de conversión una regresión exponencial, o cualquier otra función con comportamiento similar. En la figura A3-11 se presenta una fotografía del sensor. A su derecha se esquematiza el fundamento de su funcionamiento, y a su izquierda se presenta la gráfica de la relación entre la tensión de salida y la distancia junto con la formula de la función de conversión propuesta en este apéndice (los valores A y B se obtienen experimentalmente para cada uno de los sensores).

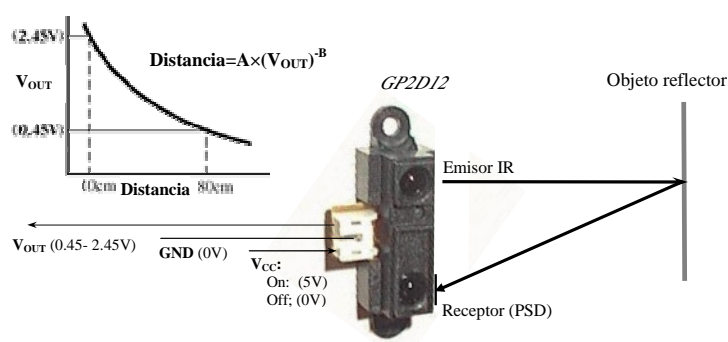


Figura A3-11. Sensor de Infrarrojos.

El valor de la señal de salida depende de la temperatura y de la iluminación del objeto reflector, aunque en el rango de temperaturas e iluminación que nos movemos el cambio no es significativo. Además, la dependencia de la tensión de salida respecto al color del elemento reflectante es despreciable. En cualquier caso, se puede considerar que el error de la tensión es menor que 200 mV, y debido a la no linealidad existente en la relación entre la tensión y la distancia, el error a cada distancia no es constante.

Aunque el *PIC16F873/6* posee un módulo de conversión A/D el número de entradas de este tipo se encuentra limitado a seis, por lo que optamos por utilizar los conversores A/D *PCF8591* de 8 bits de *Philips* (www-us.semiconductors.philips.com), controlables a través del bus I²C. Estos conversores permiten medir hasta cuatro señales analógicas de entrada, generar una de salida, definir por medio de dos pines el valor máximo (en nuestro caso 2.5 V obtenidos con un circuito divisor de tensiones) y mínimo (0 V) de las señales analógicas de entrada, y fijar, utilizando tres pines más, los tres bits menos significativos de la dirección I²C del dispositivo (1001A₂A₁A₀ para escritura y 1001A₂A₁A₀1 para lectura). Para leer las cuatro entradas analógicas después de direccionar el dispositivo para escritura hay que enviar la orden de lectura independiente de las cuatro tensiones, direccionar de nuevo el dispositivo para lectura y leer las cuatro palabras de 8 bits de cada una de las entradas. La velocidad del conversor viene determinada por la velocidad del bus I²C, realizando cada conversión con la lectura de una de las palabras. El microcontrolador encargado de solicitar las medidas cada vez que recibe a través del puerto serie una orden de lectura de infrarrojos (de las cuatro señales asociadas a un conversor A/D) es el *PIC16F873/6* responsable del control del bus I²C. Tras ejecutar la orden devolverá los valores digitales asociados a las cuatro entradas analógicas del conversor A/D elegido.

Las medidas del cinturón de infrarrojos, formado por 8 sensores distribuidos sobre la superficie octogonal del robot, de acuerdo con el esquema de la figura A3-12, son tomadas a través de dos conversores A/D, con 4 sensores por conversor. La distribución de los sensores del frente (dos) y de cada lado (otros dos), responde a la necesidad de medir la distancia a objetos cercanos cuando el robot se mueve a través de entornos estrechos. Debido a la disposición de las parejas de sensores, y

a las perturbaciones que la emisión de cada sensor del par podría inducir en las medidas del otro, las conexiones de los sensores a los conversores A/D se han realizado de forma alternada, y los sensores se encienden antes de realizar las lecturas y se apagan inmediatamente después, utilizando para este objetivo dos de las salidas digitales del *PIC²C*. Por el elevado consumo de los sensores de infrarrojos, es necesario poner una etapa de potencia formada por un conjunto de transistores en cascada entre la señal digital de encendido y apagado procedente del microcontrolador (que únicamente puede proporcionar 35 mA), y el pin de alimentación de los sensores de infrarrojos. Una vez encendido un grupo de infrarrojos, tras la recepción de una orden de lectura proveniente del PC, es necesario esperar un mínimo de 52 ms antes de que el sensor tenga el primer valor disponible, y por lo tanto, antes de comenzar la conversión A/D. La espera de este tiempo se consigue habilitando la cuenta del timer2 (configurado como temporizador con los valores necesarios para desbordar a los 52 ms) tras encender los sensores y esperando su desbordamiento antes de comunicarse con el conversor A/D correspondiente.

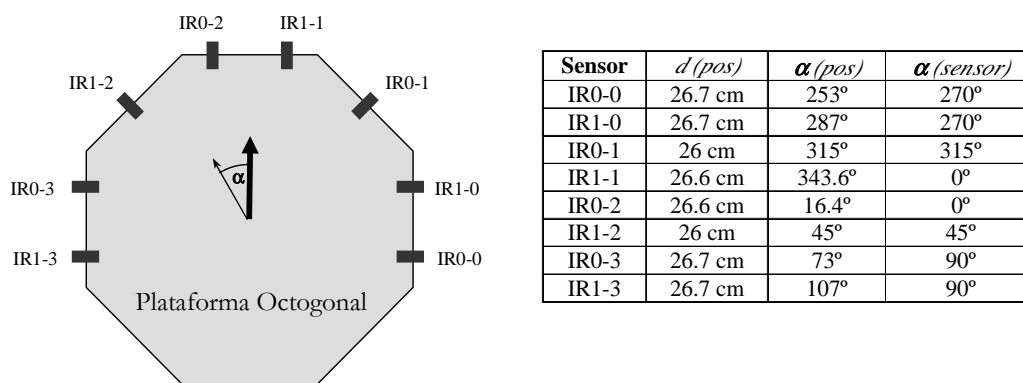


Figura A3-12. Disposición del cinturón de Infrarrojos. En la tabla $d(pos)$ es la distancia del sensor al centro del robot, $\alpha(pos)$ la orientación de la posición del sensor relativa a la orientación del robot y $\alpha(sensor)$ la orientación del sensor relativa a la orientación del robot.

El diagrama lógico de control relativo al cinturón de infrarrojos se encuentra, junto con el de la brújula, en la figura A3-10. En caso de ser necesario incluir más sensores en el cinturón de infrarrojos, bastará con replicar toda la lógica asociada a un grupo de infrarrojos, conectarla al bus I²C y a un pin configurado como salida digital del microcontrolador (de donde se obtendrá la señal de encendido y apagado) y modificar el programa del PIC para que pueda controlar el nuevo grupo de infrarrojos (encenderlo y apagarlo, y direccionar el nuevo conversor A/D).

A3.4.4. Cinturón de Ultrasonidos

El cinturón de ultrasonidos está formado por un conjunto de pares de emisores-receptores de ultrasonidos *MA40S4R/S* de *Murata* (www.murata.com) junto con sus correspondientes circuitos de excitación, filtrado y detección. Estos circuitos se han construido de acuerdo a las especificaciones de los sensores del cinturón de ultrasonidos del robot Mar IV de la Universidad de Castilla la

Mancha. Las principales características de este par emisor-receptor de ultrasonidos son su bajo peso, tamaño y consumo; su alta sensibilidad y fiabilidad (con una resolución de 9 mm); tener un rango de detectabilidad de 0.2 a 4 m cuando se enfrentan el emisor y el receptor (y de 0.1 a 2 m en el caso de que se mida la distancia a un objeto que actúe como reflector) y una directividad de 80°; y funcionar con una señal de excitación continua cuadrada con un valor máximo pico a pico de 20 V y una frecuencia nominal de 40 KHz. Su rango de detectabilidad y su directividad pueden ser mejorados (aumentando el rango hasta los 6.5 m y la directividad hasta los 30°) utilizando un concentrador de haz en ambos elementos del sensor. En la figura A3-13 se presenta una fotografía de los emisores-receptores MA40S4R/S y dos fotografías del sensor construido, una frontal donde se observan sus circuitos sin los MA40S4R/S, y otra lateral del sensor con los MA40S4R/S y los concentradores de haz.

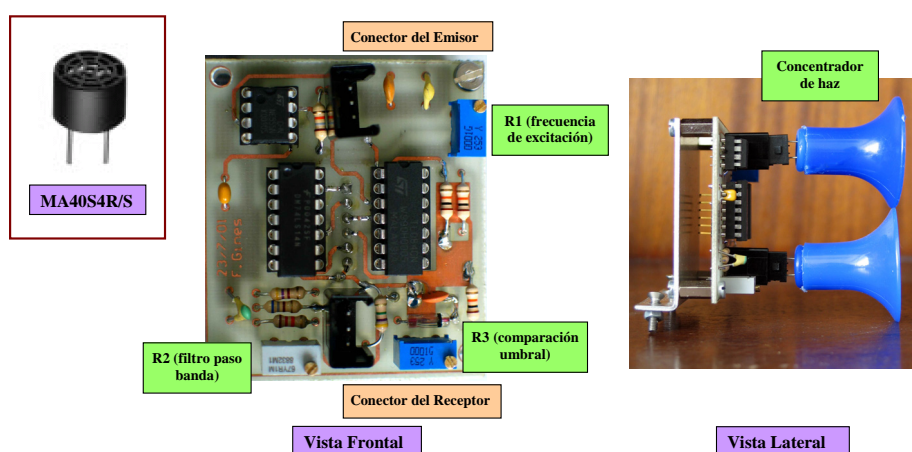


Figura A3-13. Sensor de ultrasonidos

El circuito asociado a cada par emisor-receptor, esquematizado en la figura A3-14, consta de un circuito de excitación del emisor, que a partir de una señal de emisión TTL (gráfica E1 de la figura) genera la señal eléctrica cuadrada continua de frecuencia próxima a 40 KHz responsable de que el emisor produzca ondas ultrasónicas, y de un conjunto de circuitos de filtrado y detección, responsables de amplificar y convertir la señal del receptor en una señal TTL (gráfica R3) que permita estimar el tiempo de vuelo de la señal ultrasónica emitida.

El circuito de excitación está formado por el oscilador 555, un amplificador en configuración comparadora y un conjunto de resistencias y condensadores, cuya misión es convertir una señal de emisión TTL (gráfica E1) en una señal de excitación cuadrada de ± 12 V con una frecuencia de 40 KHz (gráfica E3) mientras la señal de emisión está en alta (gráfica E2) y una señal fija de -12 V mientras la señal de emisión está en baja. Así, por medio de este circuito, se puede activar y desactivar la emisión de ultrasonidos. La frecuencia de la señal de excitación se sintoniza al valor deseado (40 KHz) por medio de la resistencia variable R1 de la figura A3-13.

Cuando el receptor recibe la señal ultrasónica reflejada por algún objeto próximo, genera una señal oscilante, de la misma frecuencia que la emitida, muy tenue, que para ser detectada es amplificada para frecuencias próximas a 40 KHz. La señal amplificada (gráfica R1) atraviesa un demodulador no coherente que obtiene la envolvente positiva de la señal. A continuación, por medio de un amplificador en configuración comparadora, se compara la envolvente con una tensión fija umbral (nivel mínimo de detección de la señal) y se obtiene la señal básica de detección (gráfica R2). Por último, utilizando un divisor de tensiones y unos negadores Trigger-Schmitt para convertir la señal a niveles TTL (gráfica R3), se obtiene la señal de recepción con la que se puede determinar el tiempo de vuelo. El filtro paso banda y la tensión umbral son sintonizados por medio de las resistencias variables R2 y R3 de la figura A3-13, lo que nos permite sintonizar el rango de funcionamiento de detección del sensor diseñado.

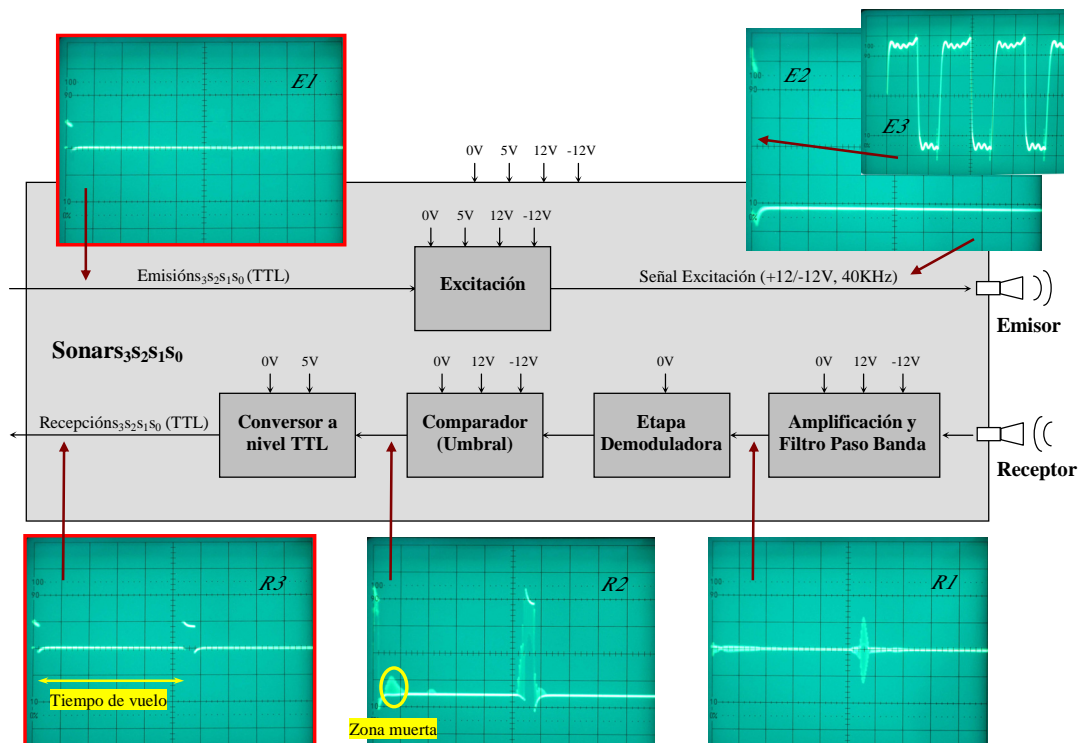


Figura A3-14. Sensor de ultrasonidos. En las gráficas la escala del eje de ordenadas es de 5 V por cuadrado, mientras que el de abscisas es de 1 ms, salvo en la gráfica E3 que es de 10 μ s.

Tras la descripción de la lógica asociada a cada sensor se presenta el funcionamiento de *PICSonar*, el microcontrolador responsable de controlar el funcionamiento de los sensores del cinturón de ultrasonidos (es decir, seleccionar el sensor, generar la señal de emisión y calcular el tiempo de vuelo) y los efectos del mismo sobre las medidas que permite obtener. En la figura A3-15 se muestra la lógica asociada al control del cinturón de ultrasonidos.

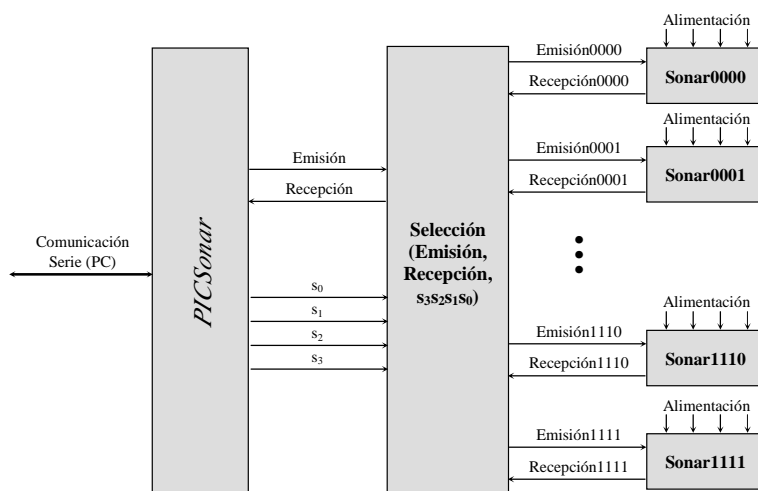


Figura A3-15. Diagrama lógico del circuito de control del cinturón de ultrasonidos.

La selección del sensor se realiza a través de cuatro pines configurados como salidas digitales y una etapa externa de selección formada por un multiplexor y demultiplexor de 16 elementos. La señal de emisión es enviada al sensor a través de un pin configurado como salida digital y el tiempo de vuelo se obtiene utilizando uno de los dos módulos de captura del microcontrolador, conectando la señal de recepción al pin asociado al módulo de captura utilizado. Para generar el pulso de emisión, una vez seleccionado el sensor, se coloca en el timer1 el complemento a uno del tiempo de emisión (200 μ s por defecto), se pone en alta la salida digital de emisión, se activa el timer y se espera a que desborde (finalice el tiempo de emisión) para poner en baja la misma salida digital. En ese momento se habilita en modo captura ante un flanco de subida el módulo captador/comparador/PWM1, y tan pronto como se observa la subida de la señal de recepción, el valor del timer1 es capturado y almacenado en dos registros. El valor almacenado en esos registros es el tiempo (en microsegundos) transcurrido entre el fin de la emisión y la detección de la señal de ultrasonidos reflejada por un objeto próximo, y nos permite estimar, añadiendo la duración del pulso de emisión, el tiempo de vuelo de la señal ultrasónica. Debido al acoplamiento entre la señal de emisión y de recepción durante el tiempo de emisión (observable en la zona inicial de las gráficas R2 y R3 de la figura A3-14), y a ruidos en el pulso de recepción (observables en zona muerta de la gráfica R2), es conveniente dejar una zona muerta para evitar la detección de dichos ruidos, por lo que en vez de activar el modo captura inmediatamente después de poner a baja la señal de emisión, se espera un tiempo (1000 μ s por defecto, que equivale a colocar a un objeto a 17 cm), calculado por medio del timer2 en configuración temporizadora con los valores apropiados, antes de activar el modo captura. El caso en el que no hay objeto reflector se encuentra contemplado en el programa del *PICSonar* mediante la comprobación sucesiva del flag de captura y del desbordamiento del timer1, fenómeno que ocurrirá a los $2^{16}-1$ microsegundos de la finalización del pulso de emisión. Con los valores por defecto de la duración del pulso de emisión y

de la zona muerta, y la capacidad de cuenta del timer1 antes de que se produzca un desbordamiento, el rango de distancias de los objetos reflectores debido al microcontrolador es de 0.20 a 11.17 m y la resolución de 0.2mm. Los valores del rango máximo y de la resolución son mejores que los que caracterizan al par emisor-receptor utilizado (2 m sin concentrador de haz y 9 mm respectivamente) por lo que el *PICSonar* únicamente limita la distancia mínima a la que se puede colocar un objeto (0.2 m). Por lo tanto, como se puede ver en la siguiente tabla, los efectos del microcontrolador se pueden considerar despreciables salvo para la distancia mínima medible.

	Distancia Mínima (m)	Distancia Máxima (m)	Resolución (mm)
Sensores (sin concentrador)	0.1	2	9
PICSonar	0.2	11.17	0.2
Conjunto Sensores y PICSonar	0.2	2	9

Tabla A3-3. Rango de funcionamiento y resolución de los sensores de ultrasonidos (sin concentrador de haz)

Al igual que ocurre con el resto de los microcontroladores, la medida es solicitada por el PC de control a través de una orden enviada por el puerto serie, que en una única palabra codifica la instrucción como medida y el sensor de ultrasonidos que se quiere utilizar. Una vez detectada la presencia de un objeto (o producido el desbordamiento del timer1 por su ausencia), devuelve el valor del tiempo (en microsegundos) de vuelo capturado en dos palabras de 8 bits (0xFF en las dos palabras en caso de desbordamiento). Además de la instrucción de medida, el PC de control puede enviar órdenes para variar el tiempo del pulso de emisión (que influye en el rango real de medida del conjunto emisor-receptor y concentrador de haz) y el tiempo de la zona muerta.

En el esquema de la siguiente figura se presenta la disposición actual de los sensores del cinturón de ultrasonidos sobre la plataforma circular superior del robot.

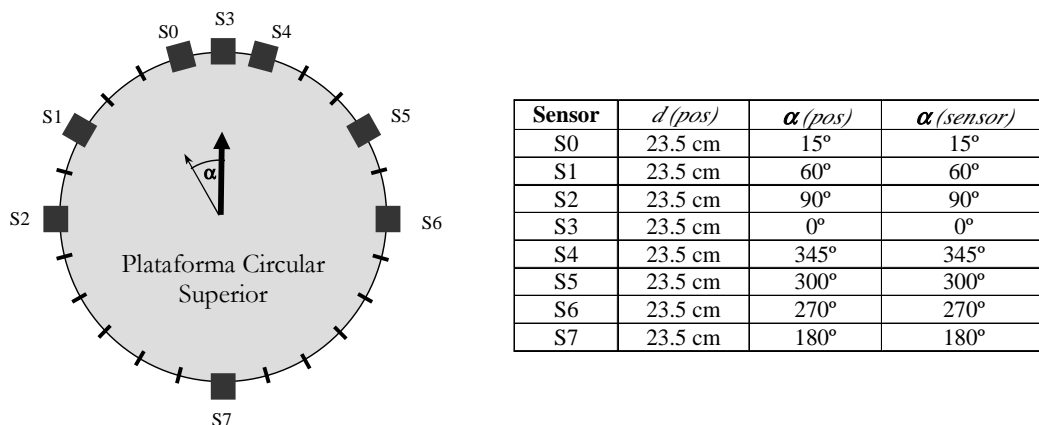


Figura A3-16. Disposición del cinturón de Ultrasonidos. En la tabla $d(pos)$ es la distancia del sensor al centro del robot, $\alpha(pos)$ la orientación de la posición del sensor relativa a la orientación del robot y $\alpha(sensor)$ la orientación del sensor relativa a la orientación del robot.

Los sensores de ultrasonidos se pueden colocar sobre dicha plataforma en diferentes posiciones, a una distancia de 23.5 cm del centro de la misma y separadas entre sí 15°. Aunque el circuito de selección permite conectar hasta 16 sensores (fácilmente ampliable a un conjunto superior modificando la etapa de selección), actualmente sólo se encuentran la mitad, distribuidos de acuerdo con el esquema de la figura A3-16. Para evitar que los ecos secundarios de unos sensores sean medidos por otros, el programa del PC de control debe evitar el acceso consecutivo a sensores próximos.

A3.4.5. Sistema de Visión

El sistema de visión del robot está formado por dos cámaras blanco y negro analógicas *XC-ST30CE* de *Sony*, colocadas sobre la plataforma octogonal, y una tarjeta de adquisición de imágenes *Matrox MeteorII/MultiChannel*, conectada al bus PCI del PC fijo y controlable desde el mismo a través de las librerías MIL de Matrox.

Las cámaras utilizadas se caracterizan por: su pequeño tamaño (44 x 29 x 57.5 mm), bajo peso (110 g) y consumo (2 W alimentadas con una señal de continua de 12 V); su elevada calidad de imagen, sensibilidad, relación S/N y tolerancia a golpes y vibraciones; y la posibilidad de proporcionarles desde el exterior una señal de sincronización de captura de imágenes que permite sincronizar el funcionamiento de ambas cámaras y obtener simultáneamente pares de imágenes de las que extraer información visual estereoscópica. La señal analógica devuelta por cada cámara se encuentra en el formato estándar CCIR, y utiliza un sistema de escaneado entrelazado de 25 imágenes por segundo. Acoplada a cada cámara hay una lente COSMICAR de 8.5 mm de longitud, cuya apertura, distancia focal, y zoom se pueden sintonizar manualmente entre los valores de F1.8 y F22, 1 m e infinito, y 12.5 y 75 mm respectivamente. El sistema, cámaras y lentes, se encuentra colocado sobre la superficie octogonal del robot, orientado hacia el frente del mismo, sobre una pequeña plataforma rectangular que permite alinear ambas cámaras según sus ejes ópticos y colocarlos a distintas distancias (17 cm en nuestros experimentos). En la figura A3-17 se presentan fotografías del sistema formado por las cámaras y las lentes y en la figura A3-18 un esquema de la localización de dicho sistema sobre la plataforma octogonal.

Las cámaras se encuentran conectadas a la tarjeta de adquisición de imágenes *Matrox MeteorII/MultiChannel*, que se caracteriza por tener hasta 6 canales de entrada seleccionables por software, a los que se pueden conectar 2 cámaras analógicas en color RGB o 6 analógicas monocromo, de las que se pueden capturar imágenes individualmente o, en el caso de tener conectadas cámaras monocromo, en grupos de 3 sincronizadas por una señal proveniente de la tarjeta (proceso equivalente a obtener las tres componentes RGB de una imagen en color). Además, tiene una salida de alimentación que permite conectar la alimentación de las cámaras al PC, y una memoria de 4 Mbytes para almacenar las imágenes temporalmente en la tarjeta, antes de enviarlas a

través del bus PCI al PC. La tarjeta puede ser controlada a través de las librerías *MIL 6.0* de *Matrox*, que operando bajo Windows (98, NT y 2000) proporcionan una amplia lista de comandos para capturar (distintos formatos standard y no standard), procesar (filtrado, operaciones morfológicas), analizar (segmentar, reconocer caracteres, reconocimiento de patrones), transferir, mostrar y archivar imágenes.

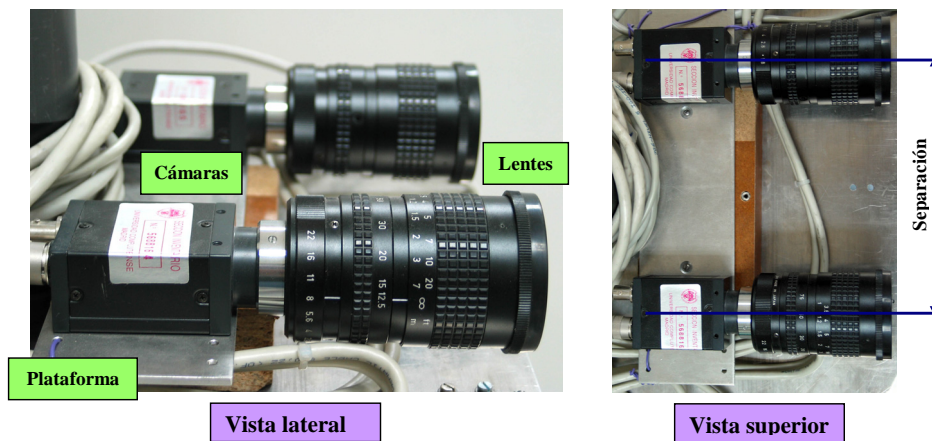


Figura A3-17. Sistema de visión: cámaras y lentes.

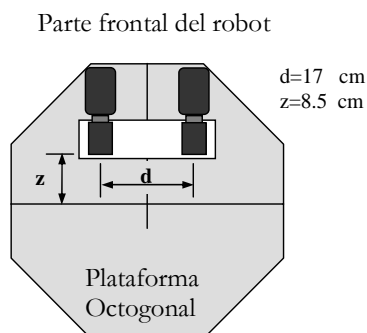


Figura A3-18. Disposición del sistema de visión.

Utilizando la tarjeta configurada como elemento de sincronización de las dos cámaras alineadas según su eje óptico y la librería MIL, se han desarrollado en el Departamento de Arquitectura de Computadores y Automática un conjunto de programas que permiten calibrar las cámaras [Vazquez2000] y obtener, como resultado de un proceso de visión estereoscópica basado en la caracterización de las regiones de las imágenes capturadas por ambas cámaras y la medida de la disparidad existente entre las mismas ([Pajares2001], [Calvache2001], [Pajares2003]), la distancia a objetos colocados delante del robot.

Estos programas han sido modificados y optimizados durante el desarrollo de la tesis para mejorar la respuesta temporal de los mismos y automatizar su funcionamiento. Además sobre las funciones del segundo programa se han construido un conjunto de librerías de C++, con dos versiones

diferentes, que permiten incorporar el proceso de visión estereoscópica a programas de control más complejos y realizar el procesamiento y análisis de las imágenes utilizando funciones de las librerías MIL (versión MIL de la librería diseñada) o funciones equivalentes desarrolladas sobre C++ (versión CAL de la librería diseñada). Esta segunda versión se desarrolló con el objeto de permitir el procesamiento y análisis de imágenes (capturadas por la versión MIL y grabadas en formato RAW en un disco duro) en PCs donde las MIL no se encuentren instaladas. Además nos permitió detectar, comparando los resultados de ambas versiones, errores de programación existentes en la versión MIL. En la versión actual de las librerías MIL cada proceso de visión estereoscópica consume unos 4 segundos, por lo que aunque la tarjeta permite capturar imágenes en tiempo real, el robot podrá extraer información visual sobre la colocación de los objetos a una velocidad máxima de 0.25 Hz.

A3.4.6. Sistema Orientable Triular de Ultrasonidos

Además de los dispositivos sensoriales ya presentados se ha diseñado y construido un sistema orientable triular de ultrasonidos, cuyo funcionamiento se encuentra explicado detalladamente en el proyecto fin de carrera [Toledano2003], que permite medir la distancia y orientación de los objetos colocados en su entorno. El sistema (figura A3-19) está construido con un emisor y dos receptores MA40S4R/S de Murata (a los que se ha asociado una lógica de excitación, filtrado y detección similares a la de los sensores de ultrasonidos descritos anteriormente) situados sobre un eje rotatorio horizontal, al que se ha acoplado un motor paso a paso ST28 de Ming Jong Technologies (www.mingjong.com.tw) controlado por el integrado SAA1027 de Philips. Un microcontrolador PIC16F873/6 (*PICSistemaTriular*) es responsable de medir los tiempos de vuelo de la señal ultrasónica capturada por los dos receptores y de enviar las órdenes de giro al controlador para que el eje se reoriente en la dirección deseada.

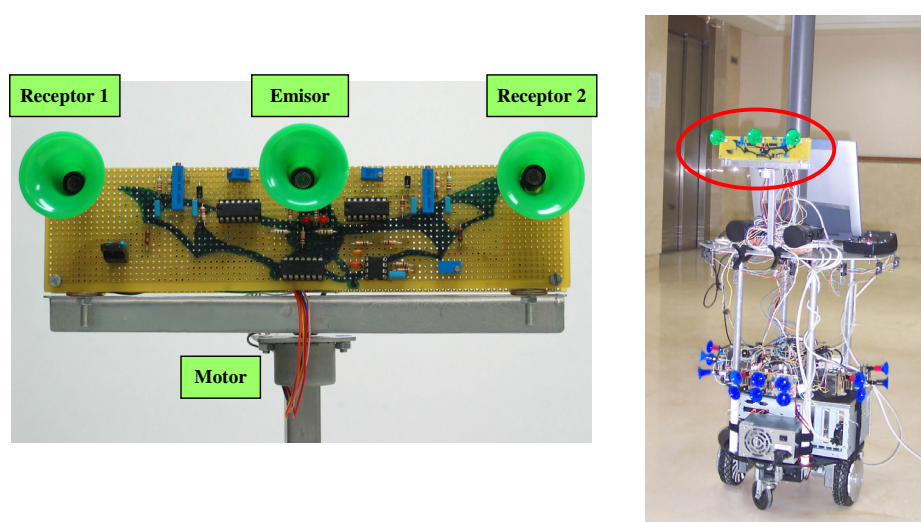


Figura A3-19. Sistema triular orientable de ultrasonidos.

El sistema, que puede funcionar de forma independiente al robot conectado a un PC a través de su puerto serie, utiliza el protocolo de comunicación serie PC-PIC del robot Nandana por lo que puede ser incorporado, cuando se desee, sobre la plataforma octogonal, dentro de la arquitectura hardware de control del mismo. El PC, como en el resto de los sistemas controlados por un *PIC16F873/6* del robot, es el responsable de solicitar al microcontrolador la medida de los tiempos de vuelo medidos por ambos receptores, a partir de los que es capaz de estimar la distancia y orientación del objeto reflector, y de indicarle el ángulo que se desea girar al eje horizontal del sistema.

A3.5. Software de Control asociado a los PIC16F873/6: Repertorio de Instrucciones

En este apartado, tras la presentación del funcionamiento de todos los dispositivos hardware del robot, se recoge el repertorio de instrucciones desarrollado para el control desde el PC de los dispositivos del robot asociado a cada uno de los microcontroladores. El repertorio diseñado para cada microcontrolador se fundamenta en el protocolo de comunicación serie PC-PIC16F873/6 distribuido, presentado en el apartado A3.2.1 de este apéndice. El PC selecciona el microcontrolador con el que desea interactuar enviando a través del puerto serie una palabra de dirección con el identificador de dicho *PIC16F873/6* y espera a recibir del mismo el identificador enviado que indica que el microcontrolador está preparado para recibir una orden. La orden está formada por tantas palabras de datos como sea necesario, empezando con una palabra que indica por cuantas palabras está formada la instrucción, y continuando con la que identifica la instrucción y toda la información adicional asociada a la misma. Tras ejecutar la orden recibida, el microcontrolador envía la respuesta dividida en un número de bytes determinado. En las siguientes tablas se recogen, para cada uno de los microcontroladores, y para cada una de las instrucciones (columna 1) de su repertorio correspondiente, las palabras de datos enviadas desde el PC (columna 2) y la respuesta enviada desde el microcontrolador (columna 3). En la palabra de datos que identifica la instrucción (la primera de la columna palabras de Datos PC->PIC para cada instrucción), se han resaltado en negrita los bits que permiten identificar las instrucciones y, para los casos en los que en esta palabra se incluya información adicional de la instrucción, en negrita cursiva los bits que la incluyen. En las tablas, *instrucción* aparece como *Ins*, *palabra menos significativa* como (- sig), *palabra más significativa* como (+ sig), *encoder derecho* como *enc der* y *encoder izquierdo* como *enc izq*.

Instrucción	Datos PC → PICMotor			Datos PICMotor → PC		
	#	Palabras	Significado	#	Palabras	Significado
Habilitar encoders	1	00000000	Identificador Ins	1	byte	OK/Error
Deshabilitar encoders	1	00010000	Identificador Ins	1	byte	OK/Error
Resetear encoders	1	00100000	Identificador Ins	1	byte	OK/Error
Leer encoders OUT: der=byte0*256+byte1 OUT: izq=byte2*256+byte3	1	00110000	Identificador Ins	4	byte0 byte1 byte2 byte3	enc der (+ sig) enc der (- sig) enc izq (+ sig) enc izq (- sig)
Leer y resetear encoders Igual que la instrucción anterior	1	00110100	Identificador Ins	4	Igual que instrucción anterior	
Leer encoders y velocidad OUT: der=byte0*256+byte1 OUT: izq=byte2*256+byte3 OUT: dir_izq=I OUT: dir_der=D OUT: vel_izq=0x7F&byte5 OUT: vel_der=0x7F&byte6	1	00111000	Identificador Ins	7	byte0 byte1 byte2 byte3 xxxxxxID byte5 byte6	enc der (+ sig) enc der (- sig) enc izq (+ sig) enc izq (- sig) Sentido rueda izq (I) y der (D) Velocidad izq. Velocidad der.
Leer encoders y velocidad, y resetear encoders Igual que la instrucción anterior	1	00111100	Identificador Ins	7	Igual que instrucción anterior	
Habilitar motores (control manual)	1	01000000	Identificador Ins	1	byte	OK/Error
Habilitar motores (control automático)	1	01001000	Identificador Ins	1	byte	OK/Error
Deshabilitar motores	1	01010000	Identificador Ins	1	byte	OK/Error
Enviar velocidad a los motores IN: dir_izq=I IN: dir_der=D IN: vel_izq (7 bits) IN: vel_der (7 bits)	3	011000/D vel_izq vel_der	Identificador Ins + Sentido rueda izq (I) y der (D) 7 bits 7 bits	1	byte	OK/Error
Enviar velocidad a los motores y leer encoders IN: dir_izq=I IN: dir_der=D IN: vel_izq (7 bits) IN: vel_der (7 bits) OUT: der=byte0*256+byte1 OUT: izq=byte2*256+byte3	3	011100/D Igual que instrucción anterior Igual que instrucción anterior	Identificador Ins + Sentido rueda izq (I) y der (D)	4	byte0 byte1 byte2 byte3	enc der (+ sig) Enc der (- sig) Enc izq (+ sig) Enc izq (- sig)
Enviar velocidad a los motores y leer y resetear encoders Igual que instrucción anterior	3	011110/D Igual que instrucción anterior Igual que instrucción anterior	Identificador Ins + Sentido rueda izq (I) y der (D)	4	Igual que instrucción anterior	

Tabla A3-4. Repertorio de instrucciones asociado a *PICMotores* (identificador 0x03).

A3.5. Software de Control asociado a los PIC16F873/6: Repertorio de Instrucciones

Instrucción	Datos PC → PIC12C			Datos PIC12C → PC		
	#	Palabras	Significado	#	Palabras	Significado
Leer orientación brújula OUT: $\alpha = (\text{byte0} * 256 + \text{byte1}) / 10$	1	01111111	Identificador Ins	2	byte0	ángulo (+sig)
					byte1	ángulo (-sig)
Medir señales del grupo de infrarrojos s0 OUT: Tensión_s0-0=byte0 OUT: Tensión_s0-1=byte1 OUT: Tensión_s0-2=byte2 OUT: Tensión_s0-3=byte3	1	00000000 _{s0}	Identificador Ins + Grupo Infrarrojos	4	byte0	Tensión sensor0
					byte1	Tensión sensor1
					byte2	Tensión sensor2
					byte3	Tensión sensor3

Tabla A3-5. Repertorio de instrucciones asociado a *PIC12C* (identificador 0x01).

Instrucción	Datos PC → PICSonar			Datos PICSonar → PC		
	#	Palabras	Significado	#	Palabras	Significado
Medir el tiempo de vuelo (μs) del sensor de ultrasonidos s3s2s1s0 OUT: $t = \text{byte1} * 256 + \text{byte0}$	1	0000 _{s3s2s1s0}	Identificador Ins + Sensor	2	byte0	tiempo (-sig)
					byte1	tiempo (+sig)
Fijar la duración del tiempo de emisión (μs) IN: temision	3	01000000	Identificador Ins	1	byte	OK/Error
		temision%128	7 bits – sig			
		temision/128	7 bits + sig			
Fijar la duración de la ventana de tiempo (μs) IN: tventana	2	01100000	Identificador Ins	1	byte	OK/Error
		tventana/40	Ventana tiempo			

Tabla A3-6. Repertorio de instrucciones asociado a *PICSonar* (identificador 0x02).

Instrucción	Datos PC → PICSistemaTriular			Datos PICSistemaTriular → PC		
	#	Palabras	Significado	#	Palabras	Significado
Medir el tiempo de vuelo (μs) del sensor OUT: $tr1 = \text{byte1} * 256 + \text{byte0}$ OUT: $tr2 = \text{byte2} * 256 + \text{byte3}$	1	01000000	Identificador Ins	4	byte0	receptor1 tiempo (-sig)
					byte1	receptor1 tiempo (+sig)
					byte2	receptor2 tiempo (+sig)
					byte3	receptor2 tiempo (-sig)
Fijar la duración del tiempo de emisión (μs) IN: temision	3	00100000	Identificador Ins	1	Byte	OK/Error
		temision%128	7 bits – sig			
		temision/128	7 bits + sig			
Fijar la duración de la ventana de tiempo (μs) IN: S _G IN: Npulsos	2	00001 _{S_G} 00	Identificador Ins + Sentido de giro	1	Byte	OK/Error
		Npulsos	Numero de pulsos			

Tabla A3-7. Repertorio de instrucciones asociado a *PICSistemaTriular* (identificador 0x04).

Los repertorios de instrucciones de control desde el PC de cada microcontrolador pueden ser programados en cualquier lenguaje de programación que permita controlar el puerto serie. En nuestro caso, hemos desarrollado una librería para Windows (NT, 2000, XP) escrita en C++ estándar, por que es el lenguaje elegido para realizar las labores de control de más alto nivel. Esta librería se encuentra compuesta por cinco clases diferentes:

- **serie_pic** es la clase básica sobre la que se implementa el protocolo de comunicación serie *PC-PIC16F873/6* para que sea usada por cada una de las clases asociadas a cada uno de los microcontroladores.
- **pic_motenc** es la clase asociada al *PICMotor*.
- **pic_i2c** es la clase asociada al *PIC²C*.
- **pic_sonar** es la clase asociada al *PICSonar*.
- **pic_torre** es la clase asociada al *PICSistemaTriular*.

Para cada una de las instrucciones de los repertorios anteriormente expuestos, se ha construido un método en la clase asociada al microcontrolador correspondiente, que no solamente está encargada de intercambiar la información, si no que también es la responsable de realizar los cálculos necesarios para devolverla en el formato más apropiado en cada caso. Por ejemplo, el método asociado a la instrucción encargada de solicitar el tiempo de vuelo en el microcontrolador *PICSistemaTriular* también calcula la distancia y ángulo del objeto reflectante por ser el objetivo final con el que ese sensor fue diseñado.

Apéndice IV

La Metodología ROOM y la Herramienta de Diseño EdROOM

En este apéndice se describen los conceptos básicos de *ROOM* (Real-Time Object Oriented Modelling), paradigma de programación gráfico orientado a objetos desarrollado por Selic, Ward y Gullecksoon [Selic1995] para el diseño, análisis e implementación de sistemas de tiempo real. Esta metodología se basa en la construcción de un modelo gráfico del sistema para definir su estructura y comportamiento, permitiendo utilizar para ambos casos tantos niveles de refinamiento como se considere oportuno. Además, ROOM contempla los problemas inherentes al diseño de sistemas de tiempo real proporcionando un conjunto de servicios, utilizables a la hora de construir el modelo, que soportan el tratamiento de este tipo de sistemas. Debido a su naturaleza recursiva y estructurada, basada en componentes que encapsulan su propio comportamiento y las relaciones existentes entre los componentes que a su vez contienen, y a su diseño orientado al dominio de tiempo real, *ROOM* es un lenguaje de programación especialmente adecuado para modelar de forma progresiva sistemas complejos de control en tiempo real.

Esta metodología, elegida para construir la arquitectura de control software de alto nivel del robot debido a su gran potencia, versatilidad y facilidad de uso, es soportada por la herramienta de distribución libre *EdROOM*, desarrollada en el Departamento de Arquitectura de Computadores y Automática de la Universidad Complutense de Madrid ([Rodriguez-Polo2003a]) y caracterizada por contemplar las tres etapas del paradigma: (1) el diseño gráfico de la estructura y comportamiento del sistema, (2) el análisis de su funcionamiento y (3) la implementación del sistema en un programa C++. El código C++ es generado de forma automática por *EdROOM* en base al modelo construido y, una vez compilado, es enlazado con una librería que proporciona los servicios básicos de la metodología ROOM. Esta librería ha sido implementada sobre varios sistemas operativos de tiempo real, lo que permitirá la portabilidad de los modelos. Así, utilizando en la etapa de enlazado la librería de servicios *EdROOM* asociada al sistema operativo elegido, el mismo modelo podrá ser ejecutado sobre diferentes sistemas operativos de tiempo real. En nuestro caso, aunque la estructura y comportamiento del programa de control es independiente de cualquier sistema operativo, a la hora de hacer uso de información sensorial visual se utiliza la versión de la librería *Mil 6.1* de *Matrox* que funciona bajo el sistema operativo Windows (98, NT o 2000), por lo que se ha desarrollado, dentro del trabajo de esta tesis, la librería de servicios *EdROOM* asociada a Windows 2000.

La metodología y herramientas elegidas para la construcción del programa de control de alto nivel del robot permiten definir, detallada y gráficamente, la estructura y comportamiento, distribuyendo la arquitectura software de control en un conjunto de tareas cooperantes, denominadas actores en ROOM. En robótica, esta división se realiza en torno a las tres etapas básicas del lazo de control (sensorización, decisión y actuación), dando lugar a tres modelos de arquitecturas software diferentes: (1) reactivas [Brooks1986] con una división en tareas (capas) responsables de las tres etapas que son capaces de sustituir (suprimir) las entradas o eliminar

(inhibir) las salidas de otras capas, (2) jerárquicas ([Albus1995], [Elfes1986]) donde la división se realiza en tareas responsables de alguna de las partes de las diferentes etapas, y (3) híbridas ([Arkin1992], [Firby1994]) cuyo objetivo es conjugar los beneficios de los dos modelos anteriores. Tanto ROOM como EdROOM permiten desarrollar de forma bien estructurada los tres tipos de arquitecturas y facilitan el diseño incremental del sistema, a través de la inclusión de nuevas tareas que aumenten su funcionalidad y/o la modificación de las ya existentes para probar nuevos algoritmos. Además, la posibilidad de generar código para diferentes sistemas operativos y de dividir el sistema de control en tantas tareas cooperantes como se desee, disminuye la dependencia existente entre las etapas de diseño e implementación y favorece el diseño de sistemas generales y reutilizables en distintos robots.

En los siguientes apartados se presentan las características más relevantes de ROOM y EdROOM, comenzando con una descripción de los aspectos de la metodología sobre los que se fundamenta nuestro sistema y la notación gráfica seguida. A continuación se describen las características de la herramienta de diseño utilizada, apartado en el que se hace un hincapié especial en el desarrollo de la librería de servicios *EdROOM* diseñada para Windows 2000. Por último, se exponen las razones principales por las que se ha elegido esta metodología y herramienta para el desarrollo e implementación de la arquitectura software de control de alto nivel de nuestro robot.

A4.1. La Metodología ROOM

ROOM es un paradigma de programación gráfica, orientado a objetos, diseñado para modelar, analizar e implementar sistemas de tiempo real, que se caracteriza por: (1) dividir el problema en elementos computacionales cooperantes llamados actores; (2) asociar a cada uno de los mismos un comportamiento propio, y (3) utilizar la cooperación de todos los comportamientos para resolver el problema. Las propiedades principales de esta metodología son:

- *Orientada a objetos*, lo que permite explotar todas las ventajas de este paradigma, como la posibilidad de reutilizar objetos previamente definidos en otros sistemas.
- *Elevado nivel de abstracción y recursividad*, como consecuencia del tratamiento de los objetos como máquinas lógicas independientes que funcionan de forma distribuida y concurrente, y la especificación de la estructura y comportamiento de dichos objetos en tantos niveles de refinamiento como se considere oportuno.
- *Específica del dominio en tiempo real*, debido al comportamiento concurrente y orientado a eventos de los objetos que constituyen el sistema, característica que permite modelar problemas complejos dentro de este dominio.
- *Cubrir todos los aspectos de desarrollo*, desde el diseño del sistema hasta su implementación final, pasando por el análisis de su comportamiento.
- *Proporcionar una documentación clara y completa de la estructura y comportamiento de todo el sistema*, extraíble de la información gráfica generada durante la fase de diseño.
- *Soportar un modelado incremental*, como consecuencia directa de todas las características anteriores, propiedad que permite incorporar de forma progresiva diferentes aspectos del sistema y verificar su correcto funcionamiento paso a paso.

La metodología ROOM ha sido utilizada para modelar el sistema de control del robot debido a estas características. En este apartado no se pretende hacer un análisis exhaustivo de la metodología, sino introducir los conceptos básicos usados en nuestro sistema y la notación gráfica de ROOM. En la figura A4-1 se esquematizan las relaciones de dependencia existentes entre los elementos de ROOM descritos a continuación.

A4.1.1. Actor

Los elementos principales de un modelo ROOM se denominan *actores* y se caracterizan por: (1) *ser máquinas lógicas* que representan una unidad funcional con un propósito definido que puede ejecutarse de forma concurrente con otros actores, (2) *abstraer y encapsular la información* que contienen de forma que no pueda ser manipulada por otros actores, y (3) utilizar únicamente el *intercambio de mensajes* para interactuar con otros actores.

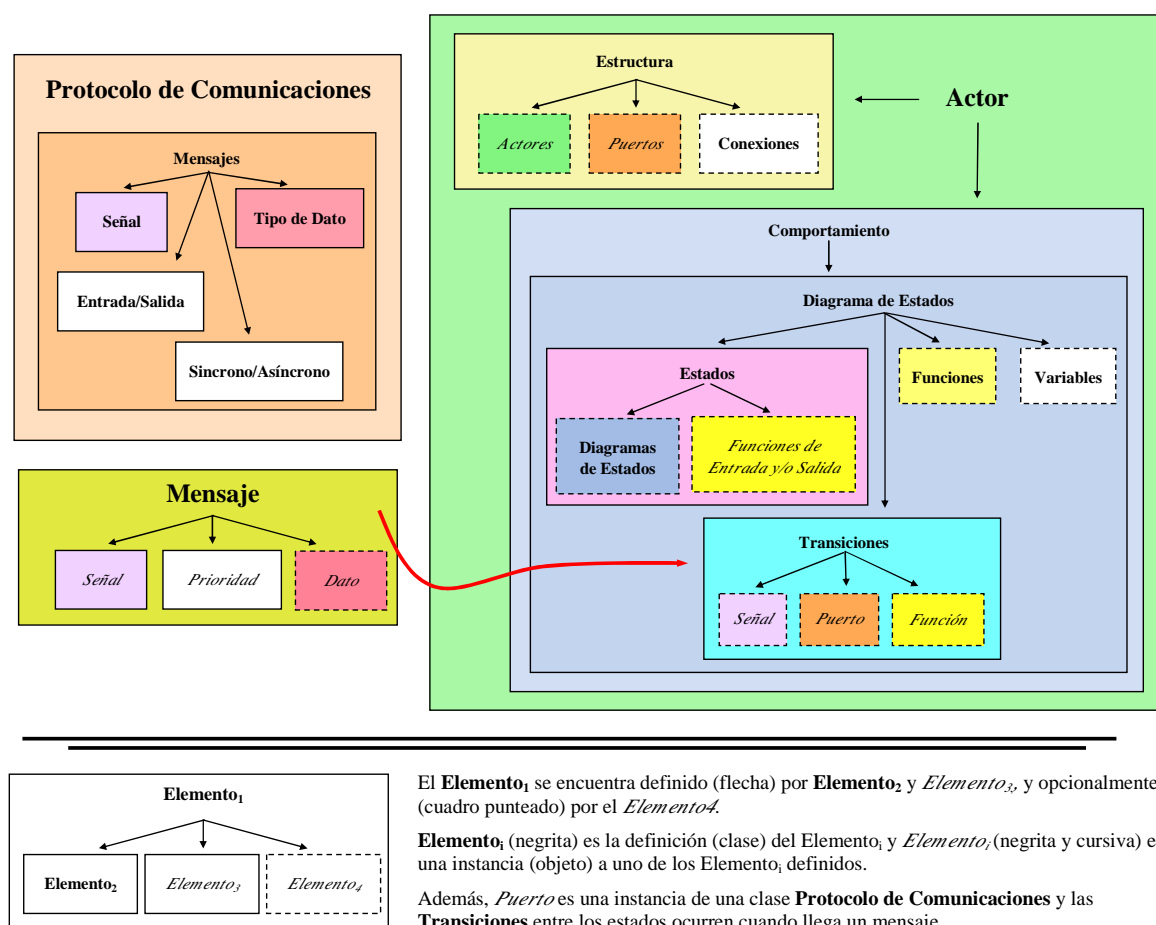


Figura A4-1. Conceptos básicos de ROOM.

La definición de un actor, representada en el esquema derecho de la figura A4-1, consta de dos partes diferenciadas:

- Su *estructura*, donde se especifican gráficamente: (1) los *puertos* de comunicación a través de los cuales intercambia información con otros actores, (2) los *actores componentes* que, en el caso de existir, contiene y (3) las *conexiones* existentes entre los puertos del actor y los de sus actores componentes.
- Su *comportamiento*, por medio de un *diagrama de estados* (basado en el formalismo *Statecharts* de Harel [Harel1987]) cuyas *transiciones* son disparadas cuando se recibe uno de los posibles mensajes asociados a las mismas, y cuya funcionalidad se concreta por medio de las *funciones asignadas a las transiciones*, y a la *entrada y salida de cada estado*.

A la hora de definir la estructura y el comportamiento se pueden utilizar varios niveles de refinamiento. En el primer caso, la estructura multinivel se consigue incluyendo, dentro del actor que se está definiendo, instancias de actores definidos previamente. En el segundo, cada estado puede encapsular a su vez otro diagrama de estado, de forma que los estados, transiciones, funciones y variables definidos dentro del nuevo *contexto* no puedan ser utilizados fuera de él. Por

lo tanto, los actores (definidos a través de su estructura y comportamiento) pueden estar formados por actores y los estados (que constituyen el diagrama de estado asignado al comportamiento) por otros diagramas de estados.

A4.1.2. Protocolos de Comunicación y Puertos

Los actores interactúan entre sí a través del mecanismo de intercambio de *mensajes*, siguiendo los patrones especificados en los *protocolos de comunicación* definidos para el modelo ROOM. En los protocolos de comunicación se definen los *tipos de mensajes* que se pueden intercambiar, especificando para cada uno de ellos: (1) la *señal* identificativa del mensaje, (2) la *clase de datos* de la información asociada al mismo, y si el mensaje es de (3) *entrada o salida* y (4) *síncrono o asíncrono*. En el esquema superior izquierdo de la figura A4-1 se representan los campos utilizados para especificar cada uno de los mensajes definidos en un protocolo de comunicación. La definición de un protocolo también determina la de su *protocolo conjugado*, que es igual a la del original pero con los mensajes de entrada y salida clasificados como mensajes de salida y entrada.

Cada *puerto*, incluido en la definición de la estructura del actor, es una instancia a uno de los protocolos de comunicación definidos en el modelo, hecho que determina completamente los mensajes que se pueden enviar a través del mismo. Además, las *conexiones* entre puertos, también incluidas en la definición de la estructura, se realizan generalmente entre *puertos conjugados*.

Existen tres clases diferentes de puertos en ROOM, soportándose sobre los mismos tres tipos de intercambio de mensajes diferentes:

- Los *puertos internos* de un actor permiten que éste intercambie mensajes con sus actores componentes y pueden ser referenciados (utilizados mediante llamadas a funciones del objeto asociado al puerto) desde su comportamiento.
- Los *puertos externos* forman parte de la interfaz exterior de un actor, permiten el intercambio de mensajes con otros actores y pueden ser referenciados desde su comportamiento.
- Los *puertos repetidores* cumplen la misión de servir de punto de conexión entre la interfaz exterior de un actor y los puertos externos y/o repetidores de sus actores componentes, y no pueden ser referenciados desde su comportamiento.

A4.1.3. Envío de Mensajes

Cuando un actor necesita interactuar con otro envía un *mensaje* a través de un puerto que se encuentre conectado con un puerto del actor receptor. Éste es un objeto de datos formado por los siguientes campos (representados en el esquema inferior izquierdo de la figura A4-1):

- La *señal*, definida en el protocolo de comunicaciones asociada al puerto por el que se envía el mensaje, que permite al actor receptor diferenciar unos mensajes de otros y así identificar la transición de estado que se debe realizar.
- La *prioridad* del mensaje que es utilizada para ordenar los mensajes recibidos de forma que cada actor trate primero aquellos más importantes y que los actores que tienen que procesar mensajes de mayor prioridad se ejecuten antes que aquellos cuyos mensajes son menos prioritarios.
- El *dato* (campo opcional) donde se envía la información asociada al mensaje.

Además, ROOM permite el envío de mensajes *síncrono* y *asíncrono*: el primer mecanismo debe ser usado cuando el actor emisor tenga que quedarse bloqueado hasta recibir una respuesta del receptor y el segundo cuando el emisor espere la recepción de todo tipo de mensajes.

A4.1.4. Servicios adicionales de ROOM

La metodología ROOM contempla la existencia de un conjunto de *servicios de comunicación, planificación y temporización* cuyo objetivo es satisfacer todos los requerimientos de diseño de un modelo con ROOM. El *servicio de comunicación* es el responsable del correcto funcionamiento del proceso de intercambio de mensajes entre actores, utilizando para este fin un conjunto de colas de recepción donde ordena los mensajes de acuerdo con su tiempo de llegada y prioridad. El *servicio de planificación* soporta la ejecución concurrente de los diferentes actores, priorizando la ejecución de aquellos que tienen que tratar mensajes más prioritarios. El *servicio de temporización* permite a los actores conocer el tiempo transcurrido desde que se inició el sistema y solicitar el envío de *mensajes de temporización* (tipo especial de mensaje de ROOM) en un instante determinado.

A4.1.5. Notación gráfica

En las siguientes figuras se introduce, por medio de ejemplos, la notación gráfica utilizada por ROOM en la definición de la estructura (figura A4-2) y comportamiento (figura A4-3) de los actores. Los ejemplos utilizados se corresponden a uno de los modelos diseñados para realizar el control de alto nivel del robot. Las descripciones de dichos ejemplos, no pretenden explicar exhaustivamente la estructura y funcionamiento del modelo utilizado, si no servir como soporte a la hora de introducir la notación gráfica utilizada por ROOM.

A la hora de especificar la estructura de un modelo ROOM se utiliza la notación gráfica de la figura A4-2. En la definición de cada clase actor, sus actores componentes (que son instancias de otras clases actores definidos dentro del mismo modelo) se representan como cajas rectangulares blancas y se identifican mediante un nombre colocado en el inferior del rectángulo. Para los puertos, se utilizan distintas representaciones, según pertenezcan al actor que se está definiendo o a sus actores

componentes. Los puertos pertenecientes a la clase actor que se está definiendo se representan mediante un cuadrado, con un círculo en su interior cuando son externos o internos, y un cuadrado en su interior cuando son repetidores. Los puertos externos y repetidores se sitúan sobre el borde exterior de la estructura de la clase actor, y los internos en su interior. Cuando se instancia un actor como componente de la definición de un actor contenedor, únicamente los puertos externos y repetidores que se especificaron durante la definición del actor componente aparecen en la definición del actor contenedor, representándose en ambos casos como un cuadrado situado, en el mismo orden y posición en el que se definieron, sobre el borde de la caja rectangular que representa al actor componente. El código de colores utilizados en la representación de todos los puertos depende únicamente de si el puerto es una instancia a un protocolo de comunicaciones o a su protocolo conjugado.

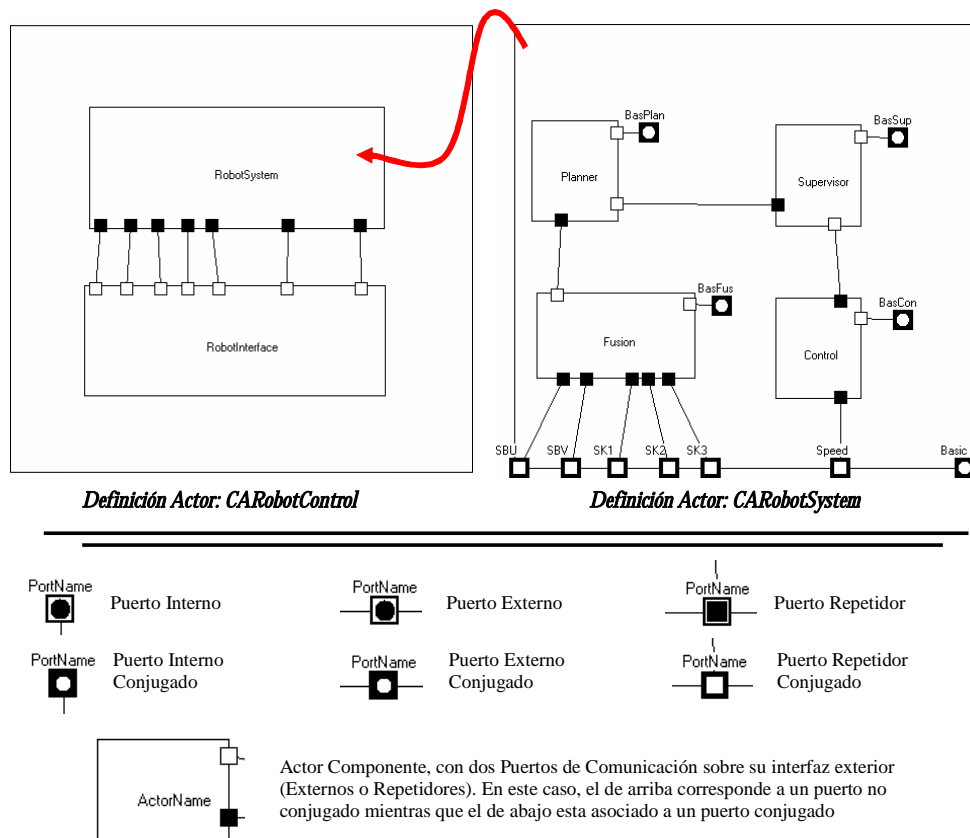


Figura A4-2. Definición de la Estructura de un Actor.

Los ejemplos elegidos para la figura A4-2 se corresponden a la definición de la estructura de dos actores, a la izquierda la de la clase actor CARobotControl y a la derecha la de la clase actor CARobotSystem. En la definición de la clase actor CARobotControl se incluyen dos actores componentes, el actor RobotSystem (instancia del actor CARobotSystem) y el actor RobotInterface (instancia a la CARobotInterface). Además, los puertos de comunicación de ambos actores componentes se encuentran conectados entre sí, y la clase actor CARobotControl no tiene ningún

puerto de comunicación propio. La clase actor CARobotSystem está formado por cuatro actores componentes (Planner, Supervisor, Fusion y Control), que se encuentran conectados entre si a través de alguno de sus puertos, y por un conjunto de puertos internos (BasPlan, BasSup, BasFus y BasCon), repetidores (SBU, SBV, SK1, SK2, SK3) y externos (Basic). Debido al encapsulamiento propio de la definición de la estructura de un actor en ROOM, aunque las clases actores de las que son instancia los actores componentes RobotInterface, Planner, Supervisor, Fusión y Control pueden estar a su vez definidos por otros actores componentes, este hecho no es visible desde la definición de las clases actores CARobotControl y CARobotSystem.

En la figura A4-3 se presenta la notación gráfica usada para especificar el comportamiento de un actor. Aunque en la misma, los estados (representados con rectángulos redondeados) y las transiciones (esquematizadas mediante flechas) aparecen únicamente identificados con una etiqueta, asociados a los mismos hay, en el caso de los estados, funciones de entrada y salida, y en el caso de las transiciones, las señales, puertos y/o acciones de guarda que las disparan y las funciones ejecutadas durante el cambio de estados.

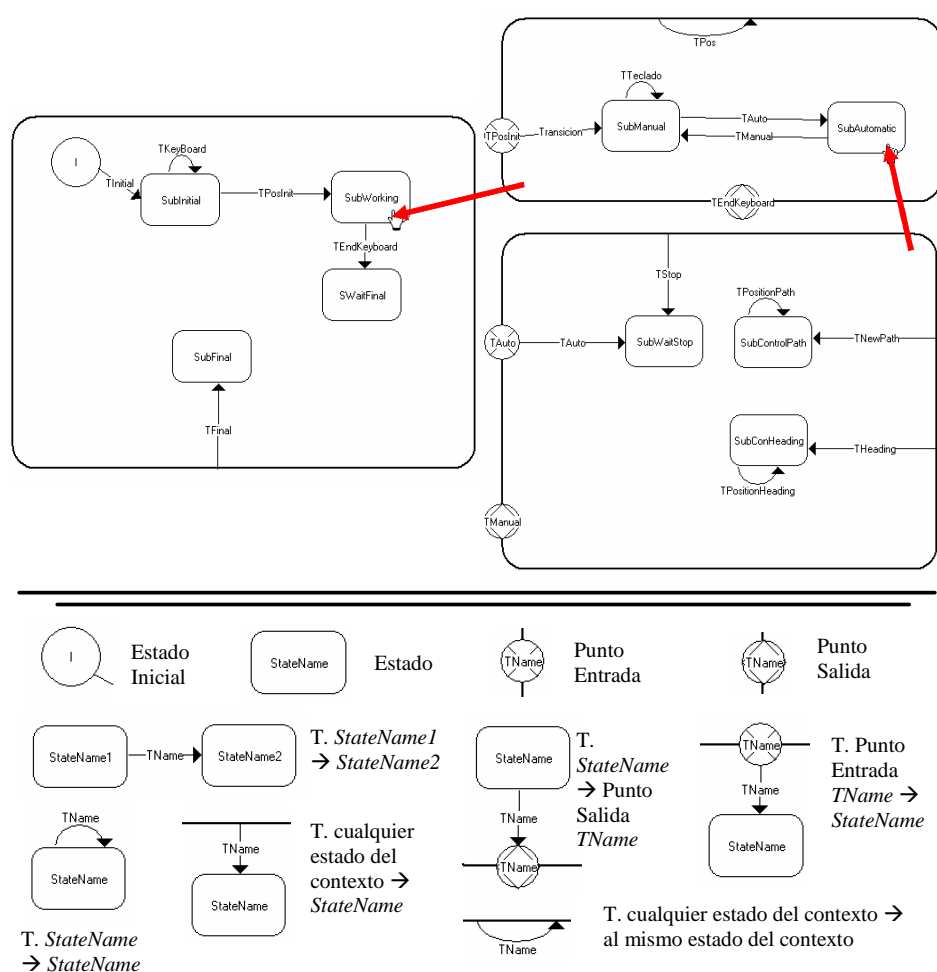


Figura A4-3. Definición del Comportamiento de un Actor.

El conjunto de diagramas de estado de la figura definen el comportamiento de la clase actor CAControl, que es instanciado como actor componente de la clase actor CARobotSystem bajo el nombre de Control. El diagrama de la izquierda, asociado al contexto superior de la clase actor CAControl, está formado por 5 estados (I, SubInitial, SubWorking, SWaitFinal, SubFinal) y determina el comportamiento de dicha clase. Los diagramas de la derecha definen el comportamiento de dos estados: el diagrama superior determina el comportamiento del estado SubWorking del diagrama de la izquierda y el diagrama inferior el comportamiento del estado SubAutomatic del diagrama superior. Cada uno de los tres diagramas define un contexto diferente, que encapsula, para contextos de nivel superior, las variables y funciones definidas dentro del mismo.

Para los contextos distintos al contexto superior, en el borde de los diagramas de estados pueden aparecer puntos de entrada (TPosIni, Tauto), representados por un círculo y una cruz, y de salida (TManual, TEndKeyboard), esquematizados por un círculo y un rombo. Los puntos de entrada y de salida son, respectivamente, puntos de llegada y salida de una transición del contexto superior, y son identificados con el nombre de las mismas. Estos puntos pueden estar unidos a los estados del contexto por medio de transiciones (Transicion, TAuto), encontrándose completamente definido dentro del contexto el estado de llegada o salida. De no estar conectados, en el caso de un punto de entrada se realizará una transición con memoria al último estado del contexto en el que se estuvo, y en el caso de un punto de salida se saldrá del estado del contexto en el que se esté. Además, las transiciones que surgen directamente desde el borde del contexto y no de un punto de entrada (TStop, TNewPath, THeading, Tpos) se ejecutan para todos sus estados, y en el caso en el que el fin de las mismas también sea un punto del borde (Tpos), el estado de entrada y salida es el mismo.

A4.2. La Herramienta de Diseño EdROOM

La metodología ROOM, que puede ser implementada de diferentes formas, es soportada por un conjunto de herramientas comerciales, como Rhapsody (www.ilogix.com), Rational Rose Realtime (www-136.ibm.com/developerworks/racional) y Statemate (www.ilogix.com), capaces de generar código a partir de la definición gráfica del sistema y caracterizadas unas por su elevado coste y otras por no proporcionar soporte para la plataforma deseada. La herramienta de diseño EdROOM, construida y presentada como tesis doctoral en el Departamento de Arquitectura de Computadores y Automática ([Rodriguez-Polo2003a]), soluciona los problemas de las herramientas comerciales al ser una herramienta de distribución libre y debido a una cuidadosa implementación que posibilita la generación de código para un numero creciente de plataformas. EdROOM es una herramienta que funciona sobre MS-Windows, formada por un editor gráfico y un generador de código que permiten definir la estructura y el comportamiento de los actores de un sistema y obtener su código C++ para distintas plataformas. Las diferentes plataformas se sustentan sobre distintas librerías de

servicios EdROOM. Entre las plataformas actualmente soportadas se encuentra el sistema operativo Windows 2000, cuya librería de servicios EdROOM ha sido desarrollado como parte del trabajo de esta tesis. En las siguientes secciones se resume el funcionamiento del editor gráfico y del generador de código, y se presenta la librería asociada a Windows 2000.

A4.2.1. Editor Gráfico

Usando el editor gráfico se pueden definir los tipos de datos, protocolos de comunicaciones y actores necesarios para construir el modelo, pudiéndose crear nuevos elementos, eliminar parte de los existentes, importarlos de otro modelo, o modificar su definición. Debido a la estructura multinivel de ROOM, es necesario definir los actores y protocolos necesarios para la definición de los actores componentes y puertos de un actor antes de que sean incluidos en la definición del mismo. A la hora de definir el comportamiento, EdROOM crea un contexto y diagrama de estados para cada actor definido y permite definir en el mismo las variables, funciones, transiciones y estados. Para cada estado se puede construir un nuevo contexto y diagrama, permitiendo tantos niveles de refinamiento como se considere oportuno. No todas las funciones utilizadas en las transiciones y entradas o salidas de los estados tienen que estar definidas dentro de los contextos, ya que EdROOM permite añadir al modelo librerías externas, destinadas a implementar código dependiente de la plataforma o cualquier código auxiliar. Además incorpora un conjunto de menús que facilitan la escritura del código relacionado con los servicios de comunicación y temporización dentro de las funciones definidas en los contextos del actor. En las dos figuras siguientes se muestran dos de las ventanas principales de EdROOM.

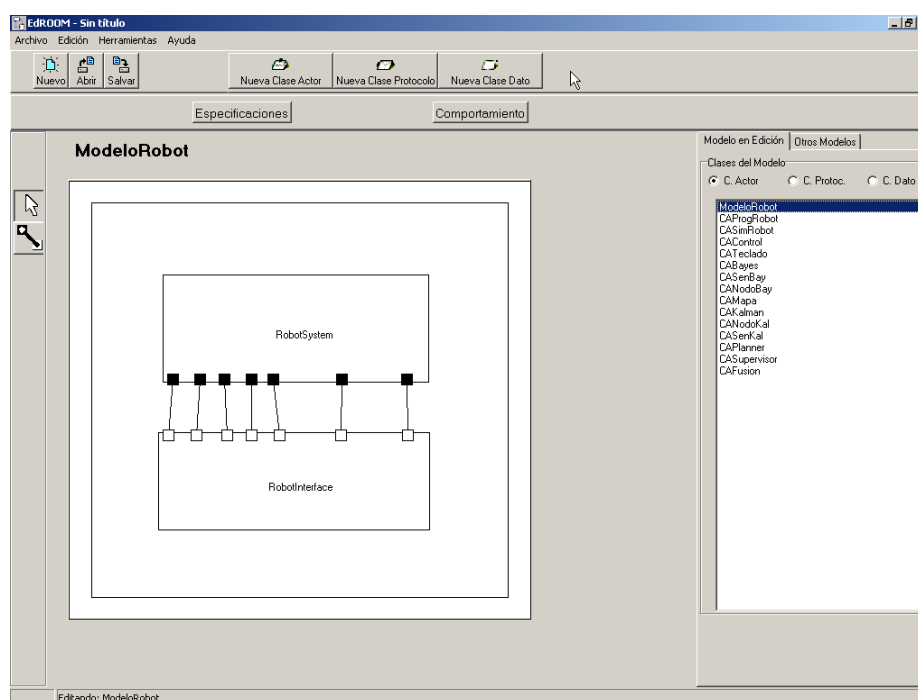


Figura A4-4. Ventana principal de EdROOM editando la estructura de CAModeloRobot.

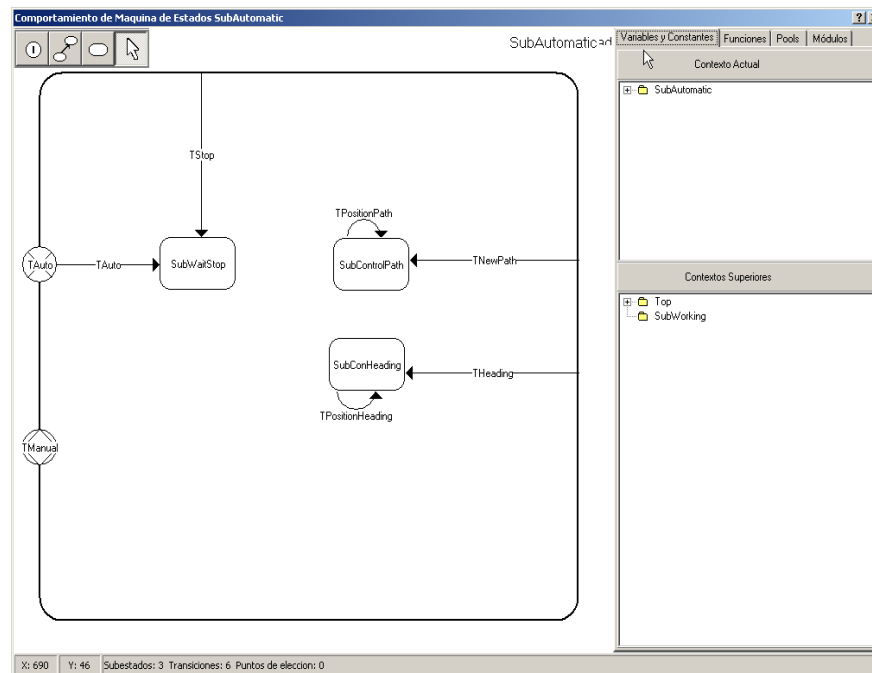


Figura A4-5. Ventana del editor de contextos de EdROOM con el estado SubAutomatic.

La figura A4-4 muestra la ventana principal del editor, con la estructura del actor CAModeloRobot a la izquierda y los actores, protocolos y clases de datos definidos a la derecha. En la figura A4-5 se presenta la ventana para definir un contexto, en este caso el relativo al estado SubAutomatic del actor CAControl, con el diagrama de estados a la izquierda, y la zona de definición de variables, funciones y módulos externos a la derecha.

A4.2.2. Generador de Código

Aunque EdROOM funciona únicamente sobre MS-Windows, el generador de código ROOM integrado en esta herramienta es capaz de generar, una vez que el modelo del sistema se encuentra totalmente especificado, código C++ para distintas plataformas de tiempo real. Además el número de plataformas puede ser fácilmente ampliable, debido a la estructura del código generado, que se esquematiza en la figura A4-6.

La estructura y comportamiento del modelo se implementan en un conjunto de clases C++ independientes de la plataforma, construidas sobre una librería de servicios EdROOM, que sirve como interfase entre el código propio del modelo y el sistema operativo de tiempo real sobre el que se desea ejecutar. Para facilitar la construcción de nuevas librerías de servicios, éstas se encuentran divididas en dos capas. La superior, también independiente de la plataforma, contiene las clases responsables de proporcionar al código del modelo los servicios ROOM de comunicación, temporización y planificación. La inferior es la responsable de implementar, haciendo uso de las funciones de la plataforma deseada, un conjunto de clases primitivas sobre las que se sustenta la capa superior. Las clases básicas de este nivel son cinco: *Pr_Task* (utilizada en la definición de

cada una de las tareas concurrentes del modelo), *Pr_Semaphore* (responsable de la sincronización y de asegurar el acceso en exclusión mutua a los recursos del sistema), *Pr_Time* (usada para encapsular el sistema de temporización del modelo), *Pr_IRQManager* (encargada del tratamiento de interrupciones), y *Pr_Kernel* (responsable del tratamiento del núcleo del modelo y de inicializar y finalizar el sistema).

Con esta estructura multicapa, a la hora de generar código para un nuevo sistema operativo sólo hay que modificar las primitivas de la capa inferior, de forma que éstas utilicen funciones propias de la plataforma deseada. Una vez construida esta capa, se compila el código de la máquina virtual y se obtiene una librería EdROOM, que el usuario enlazará con el código C++ generado para su modelo durante el proceso de compilación del mismo.

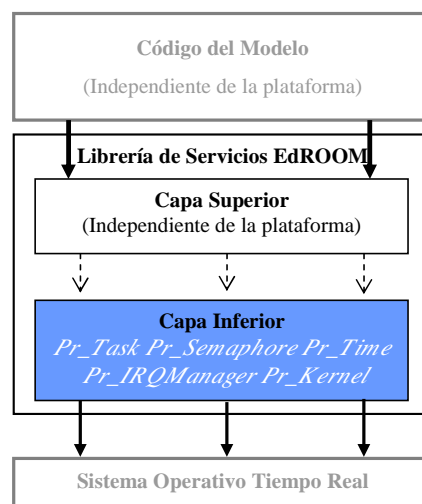


Figura A4-6. Estructura del código C++ generado por EdROOM.

Entre las plataformas actualmente soportadas por EdROOM se encuentran los sistemas operativos de tiempo real RTKernel (www.on-time.com), CMX (www.cmx.com), y RTAI (www.aero.polimi.it/~rtai/), que han sido utilizadas para el diseño de prácticas de control ([Rodríguez-Polo1998] y [Herranz-Molinero2003]), el control en tiempo real de maquetas de barcos en el Canal de Experiencias Hidrodinámicas del Pardo ([Rodríguez-Polo2001a] y [Rodríguez-Polo2001b]), y el diseño de sistemas de control de satélites ([Rodríguez-Polo2003b]).

A4.2.3. Librería EdROOM para Windows 2000

Durante el desarrollo de la tesis se ha desarrollado la librería EdROOM para Windows 2000 con el objetivo de poder modelar el sistema de control de alto nivel del robot en los casos en los que éste utilice las librerías de visión *Mil 6.1* de *Matrox*. Aunque esta librería, que puede ser utilizada por cualquier otro modelo ROOM que se desee ejecutar sobre Windows 2000, no está construida sobre un sistema operativo de tiempo real, permite construir un modelo ROOM con restricciones

temporales laxas. Además, cuando no se utilice la librería de visión, el programa de control modelado en ROOM puede ser generado para otro sistema operativo, tras sustituir los módulos externos del modelo que dependan de Windows, especificados mediante EdROOM en el comportamiento de cada actor, por otros para la plataforma deseada y enlazar el código generado por EdROOM con su librería de servicios EdROOM correspondiente.

La librería EdROOM para Windows 2000 ha sido construida para Visual C++ 6.0, por lo que las clases primitivas de la capa inferior de la máquina virtual hacen uso de funciones que este compilador proporciona para interactuar con este sistema operativo multitarea y multihilo, cuyo planificador es del tipo *Round Robin* basado en prioridades. Tras analizar las posibilidades del sistema operativo se decidió implementar cada modelo ROOM como una única aplicación Win32 de *alta prioridad* y las diferentes tareas de ROOM como hilos dentro la aplicación. Esta solución permite el uso de hasta *5 niveles de alta prioridad Win32* para los hilos, lo que asegura una mayor prioridad de la aplicación ROOM frente a la mayoría del resto de las aplicaciones Windows. Con esta configuración, sólo las aplicaciones con la prioridad *de tiempo real Win32* tendrán una prioridad siempre mayor que los *5 niveles de alta prioridad Win32* utilizados para la aplicación ROOM. Aun así, se decidió no asignar el nivel de prioridad de *tiempo real Win32* a la aplicación ROOM ya que es un nivel crítico reservado para aplicaciones con una prioridad semejante a la de los servicios propios del sistema operativo y cuyo uso indiscriminado puede provocar fallos en el mismo.

El planificador de la librería de servicios EdROOM para Windows 2000 está construido sobre el planificador de *Round Robin* basado en prioridades de este sistema operativo, lo que permite una planificación de tareas EdROOM orientada a eventos temporales. De esta forma, los comportamientos de los actores que no están a la espera de recibir un mensaje se reparten, de acuerdo con su prioridad, el tiempo de procesamiento.

Asimismo, a la hora de inicializar la aplicación, la librería EdROOM crea una consola que permite al usuario realizar las operaciones habituales de entrada y salida.

A continuación se describe brevemente cómo se han implementado las cuatro primeras clases de la máquina virtual ROOM de bajo nivel para la versión de Windows 2000 y una clase auxiliar adicional (*lista_tarea*) necesaria para mantener una lista con la información de las tareas existentes dentro de la aplicación, y se presentan las funciones de este sistema operativo que soportan dichas clases. La quinta clase, *Pr_IRQManager*, no ha sido incorporada todavía en la versión actual de la máquina virtual ROOM para Windows 2000 debido a las características de este sistema operativo. Por lo tanto, los modelos ROOM soportados por la librería asociada a Windows no pueden hacer uso de la funcionalidad propia de ROOM para el tratamiento de las interrupciones. Además, aunque la librería fue inicialmente construida para Visual C++ 6.0, también puede ser utilizada, tras

unas pequeñas modificaciones en el código del nivel inferior de la maquina virtual para el compilador C++ Builder 5.0 de Borland.

- ***Pr_task***

La clase *Pr_Task* es el soporte utilizado para definir cada una de las tareas concurrentes de ROOM, entre las que destacan las responsables de implementar el comportamiento reactivo y cooperativo de los actores del modelo. Cada tarea de la capa inferior de la maquina virtual de Windows 2000 se encuentra implementada como un hilo de la aplicación ROOM, por lo que esta clase hace uso de primitivas Windows encargadas del manejo de hilos: *_beginthread* (inicializa el hilo), *_endthread* (lo finaliza), *SetThreadPriority* (le asigna una prioridad determinada), *GetCurrentThread* (obtiene el interfaz de control del hilo) y *GetCurrentThreadId* (obtiene su identificador).

- ***Pr_Semaphore***

De la clase *Pr_Semaphore* se derivan dos clases pertenecientes también al nivel inferior de la maquina virtual, *Pr_SemaphoreBin* y *Pr_SemaphoreRec*, que implementan, respectivamente, los semáforos binarios utilizados para la sincronización y los semáforos de recursos utilizados en la capa superior. Estas clases están construidas sobre las siguientes funciones de semáforos Windows: *CreateSemaphore* (crea el semáforo), *CloseHandle* (lo elimina cuando se le pasa como parámetro de entrada un manejador de un semáforo), *ReleaseSemaphore* (lo señala) y *WaitForSingleObject* (suspende la ejecución del hilo mientras que un objeto- semáforo en este caso- no se encuentre señalado o no haya pasado el intervalo de tiempo deseado).

- ***Pr_Time***

La clase *Pr_Time* encapsula todas las funciones del sistema operativo relativas al manejo del tiempo. Aunque Windows permite usar funciones de temporización de alta resolución, a la hora de construir esta máquina virtual ROOM optamos por utilizar las funciones *GetTickCount* (devuelve el tiempo del sistema medido desde el arranque de Windows) y *Sleep* (suspende la ejecución del hilo durante el intervalo elegido). Ambas funciones utilizan los milisegundos como unidad temporal, división suficiente para esta versión de la máquina virtual de Windows ya que ésta ha sido diseñada para ser utilizada en modelos que tienen que cumplir únicamente restricciones temporales laxas. La resolución del temporizador, dependiente de la versión de Windows, es de unos 10 milisegundos para Windows 2000, característica que habrá que tener en cuenta a la hora de diseñar un modelo ROOM para este sistema operativo.

▪ ***Pr_Kernel***

La clase *Pr_Kernel* es la responsable de configurar el núcleo de la aplicación, siendo la función del mismo especialmente importante a la hora de arrancar y finalizar el sistema. Las operaciones realizadas durante estas dos etapas son fuertemente dependientes del sistema operativo sobre el que se está construyendo la capa inferior de la máquina virtual, y en el caso de Windows 2000 es la encargada de crear/destruir la consola asociada a la aplicación Win32 (utilizando una librería auxiliar especialmente diseñada para este fin) y fijar la prioridad inicial de una aplicación ROOM a uno de los 5 niveles de alta prioridad (por medio de las funciones Windows *GetCurrentProcess*, *SetPriorityClass* y *SetThreadPriority*).

▪ ***Lista_tarea***

Además de las clases anteriormente expuestas, en este nivel de la máquina virtual se ha incluido una nueva clase *lista_tarea*, responsable de almacenar en una lista la información relativa a cada una de las tareas que existen en una aplicación ROOM y de permitir el acceso a la misma a la hora de intercambiar mensajes. La lista es inicializada como un objeto estático de la clase *Pr_Kernel*, modificada por *Pr_Task* cada vez que se crea y destruye una tarea y utilizada como fuente de información por la función de la máquina virtual de bajo nivel *Pr_Receive*, que es la responsable de la recepción de los mensajes.

A4.3. Construcción de Arquitectura Software Robótica con ROOM y EdROOM

Tanto las características de la metodología ROOM como las de la herramienta de desarrollo EdROOM favorecen la construcción de una arquitectura software para el robot que sea versátil, portable y bien estructurada. Además de soportar los diferentes tipos de arquitectura robóticas (reactivas, jerárquicas e híbridas) el uso de la herramienta posibilita la adaptación de la arquitectura software diseñada a diferentes robos, sistemas operativos, y la reutilización de la misma para sistemas que incorporen nuevos algoritmos y/o funcionalidades.

En las siguientes secciones de este apartado se presentan: (1) ejemplos del uso de ROOM para construir los tres tipos de arquitectura software para robots, y (2) algunas características de la metodología ROOM y herramienta EdROOM que favorecen el diseño e implementación de la misma.

A4.3.1. Tipo de Arquitecturas Software Robóticas y ROOM

La metodología ROOM puede ser utilizada para construir los diferentes tipos de arquitecturas software robóticas, ya que es capaz de dar soporte a los comportamientos reactivos y cooperativos existentes en cada una de ellas.

La arquitectura reactiva (también conocida como descentralizada) se caracteriza por distribuir el comportamiento del sistema en un conjunto de capas que son responsables de las tres etapas básicas del lazo de control (sensorización, decisión y actuación) y que son capaces de inhibir el comportamiento de otras capas, suprimiendo la entrada a las mismas o inhibiendo sus salidas. Este tipo de arquitectura puede ser implementada bajo el formalismo ROOM identificando cada una de las capas con un actor ROOM con tres puertos de comunicación externa: uno para recibir los mensajes de sensorización, otro para enviar los mensajes de actuación, y el último para intercambiar con otros actores mensajes de activación/inhibición. En la figura A4-7 se presenta un ejemplo de la arquitectura reactiva propuesta, formado por tres niveles. El comportamiento de cada actor depende de la información proveniente de los sensores y de los mensajes de activación/inhibición procedentes de otros actores y su respuesta son mensajes con ordenes a los actuadores y/o mensajes de activación/inhibición a otros actores. La conectividad de los puertos de actuación/inhibición de todos los actores se consigue a través del actor que los contiene, responsable de recibir y reenviar este tipo de mensajes a través de sus puertos internos.

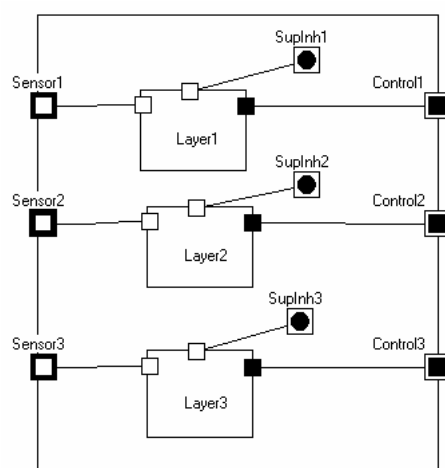


Figura A4-7. Ejemplo de arquitectura reactiva con tres capas.

En la arquitectura deliberativa (también conocida como jerárquica o centralizada) el comportamiento del sistema se segmenta en un conjunto de tareas responsables de alguna de las partes de las diferentes etapas del lazo de control. La implementación de esta arquitectura siguiendo la metodología ROOM se puede conseguir identificando las diferentes tareas con actores ROOM dotados de dos puertos: uno para el intercambio de mensajes con el actor de la etapa

anterior y el otro para el intercambio con el actor de la etapa siguiente. El comportamiento de los actores, que se encuentran conectados en serie, depende de los mensajes recibidos, soportándose de esta forma una ejecución segmentada del comportamiento del sistema. En la figura A4-8 se recoge un ejemplo de arquitectura jerárquica formada por cuatro etapas: procesamiento de la información sensorial, planificación, supervisión y control.

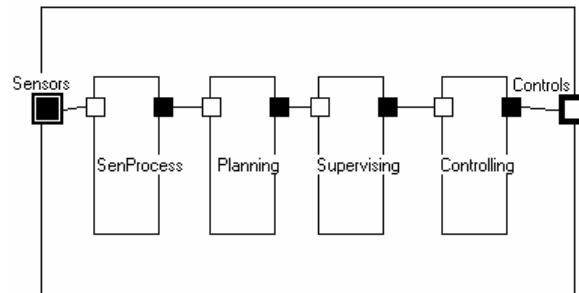


Figura A4-8. Ejemplo de arquitectura deliberativa con cuatro etapas.

La arquitectura híbrida puede ser implementada en ROOM combinando los métodos expuestos para las dos arquitecturas previamente expuestas. De hecho, la utilización de todas las características del formalismo ROOM lleva de forma natural a la implementación una arquitectura híbrida bien estructurada. En la figura A4-9 se presenta un ejemplo de arquitectura mixta, en la que el procesamiento básico se realiza en forma jerárquica, aunque se permite el envío de mensajes entre todos los actores para que unos puedan modificar el comportamiento de los otros.

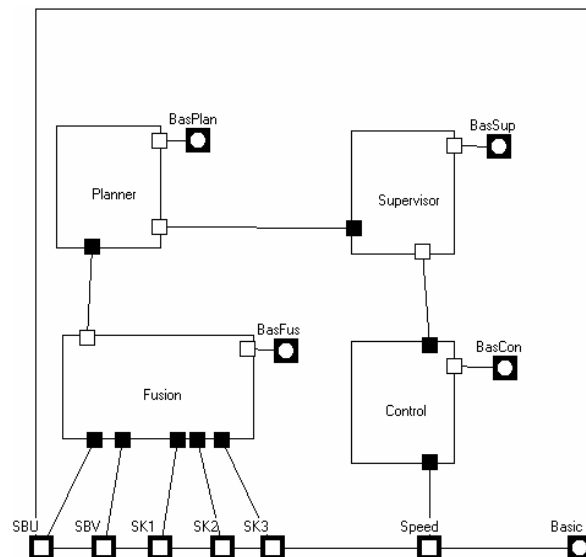


Figura A4-9. Ejemplo de arquitectura híbrida.

Por último, es importante destacar que el encapsulamiento estructural de ROOM, donde un actor puede estar compuesto por otros actores, también puede ser incorporado en los tres tipos de arquitecturas robóticas.

A4.3.2. Versatilidad de la Arquitecturas Robóticas diseñadas con ROOM y EdROOM

La definición gráfica de todo el sistema, desde su estructura hasta su comportamiento, y la existencia de tantos niveles de encapsulamiento como se considere oportuno favorece el diseño de arquitecturas software robóticas versátiles y fácilmente adaptables. Por ejemplo, una tarea de la arquitectura se puede implementar con distintos algoritmos en diferentes actores con interfaces idénticas y probarlos en el mismo sistema con una simple sustitución de actores. Otra opción es el probar el mismo sistema en diferentes robots, caracterizados por conjuntos sensoriales diferentes, siendo únicamente necesario para este caso elegir e incluir en el modelo los actores responsables de recoger y preprocesar la información sensorial propios de cada robot. Además la complejidad de la arquitectura software puede ser incrementada, siempre que sea necesaria, añadiendo nuevos actores y/o puertos de comunicación.

Aunque el paradigma ROOM puede ser implementado para una arquitectura robótica sin utilizar EdROOM [Lopez-Orozco1998b], esta herramienta gráfica facilita todas las etapas de diseño, desde el análisis del modelo hasta su implementación final. Así mismo, la versatilidad de la arquitectura se ve incrementada ante la posibilidad de generar código portable, independiente de la plataforma y del sistema operativo sobre el que va a ser usado. Únicamente es necesario disponer de la librería de servicios EdROOM para la plataforma deseada, y en caso de no existir, construirla.

Finalmente, se puede asegurar que la arquitectura robótica cumple restricciones temporales por medio del uso de servicio temporal de la maquina virtual de ROOM en aquellos casos en los que ésta ha sido construida sobre una plataforma de tiempo real. El servicio de temporización generará, en el instante que le han sido solicitados, mensajes capaces de disparar transiciones en el comportamiento de los actores. Cuando la maquina virtual se encuentra implementada sobre un sistema operativo que no es de tiempo real, como ocurre en el caso de Windows 2000, la arquitectura robótica podrá funcionar cumpliendo restricciones temporales laxas. El servicio de temporización de estas maquinas virtuales generará mensajes capaces de cumplir restricciones laxas de tiempo real y de disparar las transiciones de los comportamientos de los actores. Por lo tanto, en el caso de no necesitar que el sistema cumpla restricciones temporales duras, estas maquinas virtuales podrán satisfacer todos sus requerimientos temporales.

Apéndice V

Modelo del Sistema y de los Sensores Utilizados para Estimar la Posición del Robot Autónomo Móvil

En este apéndice se recogen las ecuaciones elegidas para modelar el movimiento del robot autónomo móvil del capítulo V y los sensores utilizados por el sistema de fusión multisensorial que forma parte de la arquitectura de control del robot para estimar con el algoritmo IFAsynMixto (algoritmo C3-3) su posición y covarianza.

La ecuación de transición de estados elegida está directamente relacionada con el sistema motriz del robot, explicado detalladamente en el apéndice III, que consta de dos ruedas motrices independientes colocadas a cada lado de la base, y dos ruedas de castor situadas en su parte anterior y posterior. Las características de este sistema permiten realizar un control de giro diferencial, y su comportamiento puede ser modelado por distintos conjuntos de ecuaciones ([Everett1995] y [Lucas2001]). El conjunto elegido para nuestro problema se fundamenta en el modelo más sencillo de los propuestos en [Lucas2001], que no tiene en cuenta la anchura de las ruedas ni las irregularidades del suelo. Descompone el movimiento del robot en dos etapas sucesivas: un desplazamiento angular debido a la diferencia de velocidad existente entre las ruedas motrices seguido de un desplazamiento lineal cuya velocidad se obtiene como el promedio de las velocidades de ambas ruedas. Esta descomposición es válida cuando los giros realizados por el robot durante su desplazamiento pueden ser considerados despreciables. El modelo de movimiento del robot, utilizado en el sistema de fusión multisensorial que forma parte de la arquitectura de control del mismo, está formado por un conjunto de ecuaciones, lineales y no lineales, que determinan la posición del robot (x , y , θ) a partir de un vector de estado formado por cinco componentes (la posición x , la posición y , la orientación θ , el desplazamiento lineal del robot Δl y el desplazamiento angular del robot $\Delta \theta$) y de la señal de control constituida por las velocidades solicitadas a la rueda izquierda y derecha (v_L y v_R respectivamente). Una característica peculiar de este sistema es que su tercer estado, la orientación θ , toma valores únicamente en el intervalo $[0, 2\pi)$, presentando una discontinuidad entre los dos valores límites del intervalo, que tiene que ser tratada adecuadamente por el algoritmo de fusión multisensorial utilizado.

Los sensores usados para estimar la posición del móvil son de tres tipos diferentes: una brújula electrónica que mide la orientación del robot, dos encoders optoelectrónicos incrementales que sirven para contabilizar el desplazamiento lineal realizado por cada una de las ruedas y cuyas medidas se encuentran relacionadas con el desplazamiento angular y lineal del robot, y ocho sensores de ultrasonidos distribuidos sobre una de sus plataformas que utilizando un mapa del entorno permiten determinar la distancia existente entre los objetos del mapa y el robot.

La brújula electrónica, que mide directamente la tercera componente del vector de estados, la orientación θ , es modelada mediante una función lineal a la que daremos un tratamiento *no lineal*

en el algoritmo IFAsynMixto debido a la discontinuidad existente en las medidas de este tipo. La brújula elegida proporciona medidas en el intervalo $[0, 2\pi)$, y para asegurar una asimilación correcta de las mismas en el espacio de información es necesario convertir el valor de la medida en un valor adecuado al valor del estado del sistema. El tratamiento no lineal es necesario cuando el estado θ fluctúa, debido a la asimilación desordenada y retardada de medidas de otros sensores, en torno a la discontinuidad existente entre 0 y 2π , ya que en ese caso es necesario recalcular el valor adecuado de la medida de la brújula y, por lo tanto, el valor de la información sensorial asociada a este sensor. En el algoritmo IFAsynMixto el recálculo de la información sensorial se producía en los sensores que el algoritmo consideraba no lineales, por lo que para asegurar que las medidas retardadas provocan el recálculo de la información asociada a las medidas de la brújula cuando el tercer estado del sistema, la orientación θ , está fluctuando en torno a la discontinuidad, el sensor será tratado como un sensor no lineal, y las medidas de la brújula serán corregidas utilizando una función especial en la etapa de asimilación.

Los encoders optoeléctricos asociados a cada una de las ruedas motrices permiten determinar el ángulo girado por las mismas y medir su desplazamiento lineal. Estas dos medidas (una por cada rueda) se encuentran relacionadas linealmente con el desplazamiento lineal y angular del robot, por lo que el modelo de medida asociado a estos dos sensores estará formado por dos ecuaciones lineales. Aun más, como los dos encoders proporcionan las medidas simultáneamente consideraremos que constituyen un único sensor cuyo modelo de medida está formado por dos ecuaciones. Este hecho constituye una centralización de las medidas proporcionadas por dos sensores antes de realizar la fusión y se fundamenta en la posibilidad de concatenar en un único vector de medida aquellas proporcionadas por múltiples sensores para realizar en el algoritmo IFAsynMixto un tratamiento monosensorial de las mismas.

Los 8 sensores de ultrasonidos distribuidos, de acuerdo con las especificaciones incluidas en el apéndice III, sobre una de las plataformas circulares del robot permiten determinar la distancia de los sensores a los objetos situados en su entorno. Cuando se dispone de un mapa del sistema es posible utilizar simultáneamente las medidas de cada uno de los sensores y el mapa para determinar la posiciones de los sensores, y mediante las ecuaciones que relacionan las posiciones de los sensores con el centro del robot, la posición de éste último. Para modelar el cálculo de la distancia existente entre los objetos del mapa y la posición de los sensores se han utilizado las ecuaciones no lineales propuestas en [Leonard1990] y [Leonard1992] para cuatro tipos de objetos diferentes: paredes, esquinas, rincones, y columnas. Las medidas de los ultrasonidos que lleguen al algoritmo IFAsynMixto y no se puedan relacionar con ninguno de los objetos del mapa serán desechadas en la etapa de validación por lo que no se usarán para determinar la posición del robot.

En las siguientes secciones se presentan las ecuaciones utilizadas por el algoritmo IFAsynMixto (C3-3) para estimar la posición del robot y el error asociado a la misma y se pone de manifiesto las modificaciones que hay que realizar en el algoritmo IFAsynMixto habitual para tratar correctamente la discontinuidad existente en los valores del tercer estado del sistema, la orientación θ , que es medido de forma directa por uno de los sensores.

A5.1. Modelo de transición de estados: ecuaciones de movimiento del robot

El sistema motriz del robot utilizado como plataforma experimental real está formado por cuatro ruedas: dos ruedas motrices controlables de forma independiente por medio de dos motores de corriente continua que se encuentran situadas a ambos lados de la plataforma inferior del robot y separadas una distancia b , y dos ruedas de castor situadas en la parte delantera y trasera de la misma plataforma encargadas de asegurar la estabilidad del robot.

Las características de este tipo de plataforma móvil permiten rotar al robot sobre si mismo y realizar un control de giro diferencial del robot en base a la velocidad de desplazamiento solicitada a cada una de las ruedas. Entre los diferentes conjuntos de ecuaciones que modelan el comportamiento de este tipo de robots ([Everett1995] y [Lucas2001]), se ha elegido un modelo sencillo, que descompone el movimiento del robot en una rotación seguida por un desplazamiento lineal, que no tiene en cuenta el grosor de las ruedas y que considera despreciables los giros que se pueden producir en el robot mientras éste se desplaza.

Para poder utilizar la señal de control de las dos ruedas en el modelo de transición de estados y los valores proporcionados por los encoders como medidas del sistema se utiliza un vector de estado formado por cinco componentes: las tres primeras (x , y , θ) determinan la posición del robot y su orientación, y las dos restantes (Δl , $\Delta\theta$) recogen los desplazamientos lineales y angulares del robot. Éstos últimos se encuentran directamente relacionados con las velocidades de control aplicadas a las ruedas (v_L , v_R). El modelo del sistema elegido, cuya función de transición de estado ideal (sin el término aditivo de ruido) se presenta en la siguiente ecuación.

$$\mathbf{x}_{t+1} = \begin{pmatrix} x(t+1) \\ y(t+1) \\ \theta(t+1) \\ \Delta l(t+1) \\ \Delta\theta(t+1) \end{pmatrix} = \begin{pmatrix} x(t) + \Delta l(t) \cos(\theta(t) + \Delta\theta(t)) \\ y(t) + \Delta l(t) \sin(\theta(t) + \Delta\theta(t)) \\ (\theta(t) + \Delta\theta(t)) \downarrow_{[0,2\pi)} \\ dt * [v_R(t) + v_L(t)]/2 \\ dt * [v_R(t) - v_L(t)]/b \end{pmatrix} \quad (\text{A5.1})$$

En las ecuaciones del modelo ideal de transición de estados, recogidas en la expresión (A5-1), se puede observar como el desplazamiento lineal del robot (Δl) se obtiene como el desplazamiento promedio de ambas ruedas (velocidad promedio multiplicada por el periodo de muestreo dt), mientras que el desplazamiento angular ($\Delta\theta$) depende de la diferencia del desplazamiento existente entre ambas ruedas (diferencia de velocidad multiplicada por el periodo de muestreo dt) y de la distancia (b) existente entre ellas. La descomposición del movimiento en un desplazamiento angular seguido de un desplazamiento lineal también puede ser observado: la nueva posición (x , y) del robot se obtiene como el desplazamiento lineal (Δl) proyectado sobre la nueva orientación del

robot (θ), calculada previamente utilizando el desplazamiento angular ($\Delta\theta$). Por último es importante destacar que tras realizar la predicción del ángulo del robot es conveniente convertir la nueva orientación calculada al rango de valores permitidos para esta componente, es decir, un valor dentro del intervalo $[0, 2\pi)$.

El jacobiano de la función de transición de estados, la matriz \mathbf{F}_t del sistema, utilizado en el IFAsynMixto como matriz de modelo de transición lineal para el cálculo de la covarianza en las etapas de predicción del algoritmo, se encuentra recogido en la expresión (A5-2).

$$\mathbf{F}_t = \begin{pmatrix} 1 & 0 & -\Delta l(t) \sin(\theta(t) + \Delta\theta(t)) & \cos(\theta(t) + \Delta\theta(t)) & -\Delta l(t) \sin(\theta(t) + \Delta\theta(t)) \\ 0 & 1 & \Delta l(t) \cos(\theta(t) + \Delta\theta(t)) & \sin(\theta(t) + \Delta\theta(t)) & \Delta l(t) \cos(\theta(t) + \Delta\theta(t)) \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{A5.2})$$

Las dos filas de ceros de la matriz \mathbf{F}_t del sistema provocan que en la etapa de predicción la nueva covarianza (calculada con la expresión $\mathbf{P}_{t+1/t} = \mathbf{F}_t \mathbf{P}_{t/t} \mathbf{F}_t^T + \mathbf{Q}_t$), los valores asociados a la predicción de la covarianza del instante de muestreo anterior ($\mathbf{F}_t \mathbf{P}_{t/t} \mathbf{F}_t^T$) sean cero en la dos últimas filas y dos últimas columnas, o lo que es lo mismo, que las covarianzas en un instante de muestreo no se propaguen a los términos de las covarianzas del siguiente instante de muestreo asociados con el desplazamiento lineal (Δl) ni angular ($\Delta\theta$).

La matriz de covarianza \mathbf{Q}_t correspondiente al término aditivo de ruido blanco de la función de transición de estados real, que se encuentra recogida en la expresión (A5-3), depende del valor del periodo de muestreo dt y del de la señal de control \mathbf{v}_t .

$$\mathbf{Q}_t(dt, \mathbf{v}_t) = dt^2 * \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{var}_{\Delta l}(v_R, v_L) & \text{cov}_{\Delta l \Delta \theta}(v_R, v_L) \\ 0 & 0 & 0 & \text{cov}_{\Delta l \Delta \theta}(v_R, v_L) & \text{var}_{\Delta \theta}(v_R, v_L) \end{pmatrix} \quad (\text{A5.3})$$

La matriz elegida únicamente tiene términos distintos de cero, en las posiciones de la matriz de covarianza asociadas con el desplazamiento lineal (Δl) y angular ($\Delta\theta$), y el valor de dichos términos depende del valor de las señales de control de cada una de las ruedas. Esta dependencia es recogida en la expresión (A5-3) a través de las funciones $\text{var}_{\Delta l}(v_R, v_L)$, $\text{var}_{\Delta \theta}(v_R, v_L)$ y $\text{cov}_{\Delta l \Delta \theta}(v_R, v_L)$. De esta forma, en la etapa de predicción, únicamente se incrementa el valor de la covarianza (por medio del término \mathbf{Q}_t) en los términos directamente asociados a la señal de control. Esta elección presupone que en el modelo real de transición de estado únicamente existen términos aditivos de ruido en las expresiones correspondientes al cálculo del desplazamiento lineal (Δl) y angular ($\Delta\theta$),

o lo que es lo mismo, que el cálculo de los tres valores de estado restantes (x , y , θ) es una operación determinista.

Debido a la forma de la matriz de transición de estados lineal F_t y de la matriz de covarianza Q_t elegida es necesario que el valor de los términos diagonales no nulos de Q_t tomen siempre valores distinto de cero, incluso cuando la señal de control del robot aplicada a las ruedas sea nula (o se encuentre en la zona muerta del robot) para asegurar la invertibilidad de la matriz de covarianza $P_{t+1|t}$ obtenida tras la etapa de predicción. Este hecho hace que el error de los estados que determinan la posición del robot (x , y , θ) aumente en la etapa de predicción aún cuando sepamos que por el valor de la señal de control elegida el robot se encuentra parado. Para evitar un crecimiento excesivo de la covarianza de esos estados cuando la señal de control elegida impida el movimiento del robot, los valores de las componentes no nulas de Q_t elegidos en ese caso serán próximos a cero.

A5.2. Modelos de medida: ecuaciones de los sensores utilizados para determinar la posición del robot

La plataforma robótica móvil elegida para probar el funcionamiento del algoritmo IFAsynMixto en un sistema real está dotada con diferentes sensores que pueden ser utilizados para determinar la posición del robot. Los sensores elegidos para este fin son una brújula electrónica que proporciona información sobre la orientación del robot, dos encoders optoeléctricos incrementales que permiten medir el desplazamiento lineal realizado por cada una de las ruedas, y ocho sensores de ultrasonidos, distribuidos sobre una de las plataformas del robot, que junto con un mapa del entorno, sirven para determinar la distancia de los sensores a los objetos del mapa, y de esta forma, proporcionan información sobre la posición del robot en su entorno. En las secciones siguientes se presentan los modelos de medida de cada uno de los sensores y se pone de manifiesto algunas peculiaridades que hay que tener en cuenta a la hora de utilizar las medidas proporcionadas por los mismos en el algoritmo IFAsynMixto.

A5.2.1. Brújula Electrónica

La brújula electrónica utilizada en el robot, cuyo funcionamiento se describe en el apéndice III, proporciona medidas, sobre el tercer estado del robot, la orientación θ , cuyos valores se encuentran definidos en el intervalo $[0, 2\pi)$. Como los intervalos de los valores de medida de la brújula y del tercer estado del sistema son el mismo, la relación existente entre la medida y el estado es la siguiente:

$$\mathbf{z}_{l,t+1} = z_\theta(t+1) = \theta(t+1) \quad (\text{A5.4})$$

El modelo del sistema es lineal, y las matrices de medida $\mathbf{H}_{l,t+1}$ y covarianza $\mathbf{R}_{l,t+1}$ son:

$$\mathbf{H}_{l,t+1} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (\text{A5.5})$$

$$\mathbf{R}_{l,t+1} = \begin{pmatrix} \text{var}_{\text{brújula}} \end{pmatrix} \quad (\text{A5.6})$$

A pesar de la linealidad del sistema, la discontinuidad existente entre los valores permitidos del tercer estado y de la medida de la brújula, que se encuentra situada en los límites del intervalo de valores permitidos, hace necesario realizar un tratamiento especial de este sensor a la hora de calcular la innovación de la etapa de validación y la información asociada a la medida realizada en la etapa de asimilación. Aun más, también hay que recalcular la información sensorial de la brújula cuando se asimilen medidas retardadas de otros sensores y el valor del tercer estado se encuentre fluctuando en torno a la discontinuidad.

El término de la innovación ($\hat{\mathbf{r}}_{s,t+1/t} = \zeta_{s,t+1} - h_s(\hat{\mathbf{x}}_{t+1/t}, t+1)$) de la etapa de validación calcula la diferencia entre el valor predicho y el medido. Al calcular la diferencia entre dos valores angulares hay que tener en cuenta la existencia de la discontinuidad y los dos caminos por los que es posible calcular la diferencia, que se encuentran ilustrados en la figura A5-1. La diferencia que interesa para el cálculo de la innovación es la del camino más corto, cuyo valor absoluto máximo siempre es menor o igual que π radianes. Sin embargo, con la operación habitual del cálculo de la innovación, y debido a la discontinuidad entre 0 y 2π , el valor absoluto de la diferencia no siempre es menor o igual que π radianes, por lo que será necesario implementar una operación de innovación para este sensor que asegure que siempre se toma la diferencia por el camino mas corto.

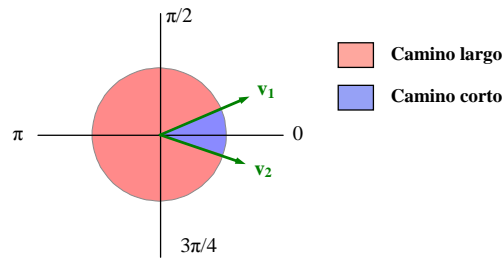


Figura A5-1. Caminos para encontrar la diferencia entre los dos ángulos definidos por los vectores \mathbf{v}_1 y \mathbf{v}_2 .

El problema de la discontinuidad también aparece a la hora de realizar la asimilación de la medida angular en el espacio de información. Cuando se utiliza la expresión habitual para un sensor lineal ($\mathbf{P}_{t+1/t+1}^{-1} \hat{\mathbf{x}}_{t+1/t+1} = \mathbf{P}_{t+1/t}^{-1} \hat{\mathbf{x}}_{t+1/t} + \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}$) y en el camino más corto entre los ángulos de $\hat{\mathbf{x}}_{t+1/t}$ y $\zeta_{s,t+1}$ se encuentra la discontinuidad, la operación, que puede entenderse como una media ponderada con las covarianzas, producirá resultados no válidos en el tercer estado.

Para ilustrar este problema, supongamos que el vector de estado únicamente está formado por el estado angular, que las covarianza $\mathbf{P}_{t+1/t}$ y $\mathbf{R}_{s,t+1}$ toman el mismo valor (igual a \mathbf{cov}), y que el valor del estado es de 359.9 grados mientras que el de la medida es de 0.1 grados. Para calcular el nuevo valor del estado tras la asimilación, es necesario calcular previamente el nuevo valor de la covarianza (con la expresión $\mathbf{P}_{t+1/t+1}^{-1} = \mathbf{P}_{t+1/t}^{-1} + \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1}$). Las operaciones y su resultado final se presentan a continuación:

$$\begin{aligned}\mathbf{P}_{t+1/t+1} &= (\mathbf{P}_{t+1/t}^{-1} + \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1})^{-1} = (\mathbf{cov}^{-1} + \mathbf{cov}^{-1})^{-1} = \mathbf{cov}/2 \\ \hat{\mathbf{x}}_{t+1/t+1} &= \mathbf{P}_{t+1/t+1} (\mathbf{P}_{t+1/t}^{-1} \hat{\mathbf{x}}_{t+1/t} + \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}) = \mathbf{cov}/2 * (\mathbf{cov}^{-1} \hat{\mathbf{x}}_{t+1/t} + \mathbf{cov}^{-1} \zeta_{s,t+1}) = \\ &= (\hat{\mathbf{x}}_{t+1/t} + \zeta_{s,t+1})/2 = (359.9^\circ + 0.1^\circ)/2 = 180.0^\circ\end{aligned}$$

El resultado obtenido es obviamente erróneo, ya que el que cabría esperar, de no existir el problema de la discontinuidad sería 0.0°. Por lo tanto, es necesario realizar algunas modificaciones en la operación de asimilación, para asegurar que las medidas angulares se asimilan de forma correcta. La solución utilizada consiste en modificar (corregir) el valor de la medida *siempre y cuando la discontinuidad se encuentre en el camino más corto entre los dos ángulos*, de forma que la medida tome un valor adecuado para ser asimilado con el valor del estado angular:

- Cuando el estado angular se encuentre en el intervalo $[\pi, 2\pi)$, como le ocurre al ángulo definido por vector \mathbf{v}_2 de la figura A5-1, al valor de la medida, ángulo definido por el vector \mathbf{v}_1 , se le suman 2π radianes.
- Cuando el estado angular se encuentre en el intervalo $[0, \pi)$, como le ocurre al ángulo definido por vector \mathbf{v}_1 , al valor de la medida, ángulo definido por el vector \mathbf{v}_2 , se le restan 2π radianes. Aunque el valor corregido de la medida en este caso será negativo, la asimilación se realizará de forma correcta.

Cuando la discontinuidad no se encuentra en el camino más corto entre los dos ángulos, no se realiza ninguna corrección.

La medida modificada $\zeta_{s,t+1}^M$, que es igual a la medida cuando la discontinuidad no se encuentra en el camino más corto entre el ángulo del estado y de medida, se utiliza entonces para calcular el valor de su información sensorial ($\mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}^M$). El nuevo estado angular obtenido tras la operación de asimilación debe ser finalmente llevado al rango de valores angulares válido, el intervalo $[0, 2\pi)$. En las operaciones de los ejemplos de la tabla A5-1 (en las que se supone que el vector de estado está sólo formado por el estado angular y que las covarianzas $\mathbf{P}_{t+1/t}$ y $\mathbf{R}_{s,t+1}$ toman el mismo valor (igual a \mathbf{cov}) por lo que $\mathbf{P}_{t+1/t+1} = (\mathbf{P}_{t+1/t}^{-1} + \mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \mathbf{H}_{s,t+1})^{-1} = (\mathbf{cov}^{-1} + \mathbf{cov}^{-1})^{-1} = \mathbf{cov}/2$) se puede observar como la corrección propuesta produce los resultados deseados.

Operaciones cuando la discontinuidad $0-2\pi$ se encuentra en el camino mas corto entre la Medida $\zeta_{s,t+I}$ y el Estado $\hat{\mathbf{x}}_{t+I/t}$	
Estado en el intervalo $[\pi, 2\pi)$ $\zeta_{s,t+I} = 0.3^\circ$ $\hat{\mathbf{x}}_{t+I/t} = 359.9^\circ$	$\zeta_{s,t+I}^M = (0.3^\circ + 360.0^\circ) = 360.3^\circ$ $\hat{\mathbf{x}}_{t+I/t+I} = \mathbf{P}_{t+I/t+I} (\mathbf{P}_{t+I/t}^{-1} \hat{\mathbf{x}}_{t+I/t} + \mathbf{H}_{s,t+I}^T \mathbf{R}_{s,t+I}^{-1} \zeta_{s,t+I}^M) = \text{cov}/2 * (\text{cov}^{-1} \hat{\mathbf{x}}_{t+I/t} + \text{cov}^{-1} \zeta_{s,t+I}^M) =$ $= (\hat{\mathbf{x}}_{t+I/t} + \zeta_{s,t+I})/2 = (359.9^\circ + 360.3^\circ)/2 = 360.1^\circ$ $\hat{\mathbf{x}}_{t+I/t+I} = \hat{\mathbf{x}}_{t+I/t+I} \perp_{[0,2\pi)} = 360.1^\circ \perp_{[0,2\pi)} = 0.1^\circ$
Estado en el intervalo $[0, \pi)$ $\zeta_{s,t+I} = 359.7^\circ$ $\hat{\mathbf{x}}_{t+I/t} = 0.1^\circ$	$\zeta_{s,t+I}^M = (359.7^\circ - 360.0^\circ) = -0.3^\circ$ $\hat{\mathbf{x}}_{t+I/t+I} = \mathbf{P}_{t+I/t+I} (\mathbf{P}_{t+I/t}^{-1} \hat{\mathbf{x}}_{t+I/t} + \mathbf{H}_{s,t+I}^T \mathbf{R}_{s,t+I}^{-1} \zeta_{s,t+I}^M) = \text{cov}/2 * (\text{cov}^{-1} \hat{\mathbf{x}}_{t+I/t} + \text{cov}^{-1} \zeta_{s,t+I}^M) =$ $= (\hat{\mathbf{x}}_{t+I/t} + \zeta_{s,t+I})/2 = (0.1^\circ - 0.3^\circ)/2 = -0.1^\circ$ $\hat{\mathbf{x}}_{t+I/t+I} = \hat{\mathbf{x}}_{t+I/t+I} \perp_{[0,2\pi)} = -0.1^\circ \perp_{[0,2\pi)} = 359.9^\circ$

Tabla A5-1. Ejemplos de las operaciones realizadas con las medidas corregidas cuando la discontinuidad $0-2\pi$ se encuentra en el camino más corto entre la medida y el estado.

En los casos en los que la medida de la brújula $\zeta_{s,t+I}^M$ tiene que ser modificada conforme al valor del estado angular antes de realizar la asimilación, la información sensorial asociada a este sensor ($\mathbf{H}_{s,t+I}^T \mathbf{R}_{s,t+I}^{-1} \zeta_{s,t+I}^M$) dependerá del estado. Por lo tanto, la asimilación retardada de medidas provenientes de otros sensores que modifiquen el valor del estado angular y lo muevan al otro lado de la discontinuidad hará que la información calculada para la brújula deje de ser correcta, y por lo tanto, que deba ser recalculada.

La necesidad del recálculo de la información sensorial ya se encuentra contemplada en el algoritmo IFAsynMixto para todos los sensores no lineales, hecho que nos permite adaptar el algoritmo general a nuestro problema particular con tan sólo definir la brújula como un sensor no lineal y sustituir para este sensor la operación general de la corrección de medida ($\zeta_{s,t+I}^C = \zeta_{s,t+I} - h_s(\hat{\mathbf{x}}_{t+I/t}, t+I) + \mathbf{H}_{s,t+I} \hat{\mathbf{x}}_{t+I/t}$) por la operación de corrección explicada previamente.

El tratamiento no lineal de la brújula durante el resto del algoritmo no influirá en los resultados obtenidos, ya que el hecho de tratar un sensor lineal como no lineal no modifica el resultado final, aunque si aumenta la necesidad de memoria (necesaria para almacenar la medida por si hay que recalculer la información sensorial) y el número de operaciones (cuando se produce el recálculo de la información sensorial). Estos aumentos son necesarios por si se produjesen las fluctuaciones del valor del estado en torno a la discontinuidad, por lo que el tratamiento no lineal del sensor se encuentra totalmente justificado. Aun más, el uso de medidas angulares y los problemas que surgen con la discontinuidad impiden la utilización de IFAsynMixtoEsp (la versión modificada del algoritmo IFAsynMixto), ya que ésta se caracterizaba por ahorrar el recálculo de la información sensorial, y este cálculo es crucial para la información de la brújula en los casos en los que el estado angular fluctúe en torno a la discontinuidad.

Por lo tanto, para el correcto funcionamiento del algoritmo IFAsynMixto con medidas angulares procedentes de la brújula, además de definir el modelo de medida (A5-4), su jacobiano (A5-5) y su matriz de covarianza (A5-6), se han implementado y utilizan unas funciones de cálculo de la innovación y de corrección de medida especiales para este sensor, y a pesar de la linealidad por tramos continuos propia del modelo, la brújula es tratada como un sensor no lineal para evitar los problemas que surgen con la discontinuidad y la asimilación retardada de medidas procedentes de otros sensores.

A5.2.2. Encoders ópticos incrementales

Los encoders ópticos incrementales utilizados en el robot, cuyo funcionamiento se describe en el apéndice III, proporcionan medidas sobre el desplazamiento de cada una de las ruedas (N_L desplazamiento de la rueda izquierda y N_R de la derecha). Cuando los instantes de muestreo y de la aplicación de la señal de control son iguales, estos desplazamientos se encuentran relacionados con los estados cuarto y quinto, el desplazamiento lineal (Δl) y angular del robot ($\Delta \theta$), y pueden ser modelados por las funciones lineales recogidas en la siguiente ecuación, en las que b representa la distancia existente entre las ruedas motrices:

$$\mathbf{z}_{2,t+1} = \begin{pmatrix} z_{N_L}(t+1) \\ z_{N_R}(t+1) \end{pmatrix} = \begin{pmatrix} \Delta l(t+1) - \Delta \theta(t+1) * b/2 \\ \Delta l(t+1) + \Delta \theta(t+1) * b/2 \end{pmatrix} \quad (\text{A5.7})$$

El modelo del sistema es lineal, y las matrices de medida $\mathbf{H}_{2,t+1}$ y covarianza $\mathbf{R}_{2,t+1}$ son:

$$\mathbf{H}_{2,t+1} = \begin{pmatrix} 0 & 0 & 0 & 1 & -b/2 \\ 0 & 0 & 0 & 1 & b/2 \end{pmatrix} \quad (\text{A5.8})$$

$$\mathbf{R}_{2,t+1}(dt) = dt^2 \begin{pmatrix} \text{var}_{N_L} & 0 \\ 0 & \text{var}_{N_R} \end{pmatrix} \quad (\text{A5.9})$$

Hay que tener en cuenta que a pesar de tratarse de dos sensores que proporcionan medidas independientes entre si (de ahí que a las covarianzas cruzadas de la matriz $\mathbf{R}_{2,t+1}$ se les haya asignado un valor nulo), éstas tienen que ser tomadas en el mismo instante de muestreo para poder ser relacionadas con las dos últimas componentes del vector de estados del sistema por lo que han sido concatenadas en un único vector de medida para tratar la información sensorial odométrica de forma conjunta. Aun más, *los instantes de lectura de los encoders y de aplicación de la señal de control deben coincidir, para que las medidas puedan ser asimiladas correctamente.*

A5.2.3. Cinturón de ultrasonidos y mapa del entorno

El cinturón de ultrasonidos, formado por 8 sensores cuya distribución sobre una de las plataformas circulares del robot se encuentra detallada en el apéndice III, proporciona medidas sobre la

distancia de los objetos que se encuentran en un entorno próximo al robot. Estas medidas de distancia sirven para posicionar al robot en el espacio cuando se conoce la posición de los objetos del entorno. El modelo elegido para determinar la posición del robot a partir del mapa se fundamenta en el propuesto en [Leonard1990] y [Leonard1992], en el que se supone que el mapa está únicamente formado por cuatro tipos de objetos: superficies planas, esquinas, rincones, y superficies cilíndricas. El modelo elegido propone una ecuación diferente para los distintos tipos de elementos del mapa:

- La distancia entre una pared y el sensor se encuentra determinada por la ecuación de distancia entre un punto (que determina la posición del sensor $(x_s(t+I), y_s(t+I))$) y una recta (la que define la pared en el plano y se encuentra representada como $A_{obj}x + B_{obj}y + C_{obj} = 0$, con $A_{obj}^2 + B_{obj}^2 = 1$), siempre y cuando la orientación relativa entre el sensor y la ortogonal a la pared no supere un determinado umbral, relacionado con la directividad del sensor.
- La distancia entre un rincón o una esquina y el sensor se encuentra determinada por la ecuación de distancia entre dos puntos (que determinan las posiciones del objeto (x_{obj}, y_{obj}) y del sensor $(x_s(t+I), y_s(t+I))$), siempre y cuando la diferencia entre las orientaciones del sensor y de la recta que une los dos puntos no supere un determinado umbral, relacionado con la directividad del sensor.
- La distancia entre una columna y el sensor se encuentra determinada por la ecuación de distancia entre un punto (que determina la posición $(x_s(t+I), y_s(t+I))$) y un círculo (que determina la posición (x_{obj}, y_{obj}) y radio r_{obj} del objeto), siempre y cuando las orientaciones del sensor y de la recta que une el sensor con el centro de la columna no superen un determinado umbral, relacionado con la directividad del sensor.

Como el objeto del uso del sensor es determinar la posición del robot y no la de sus sensores de ultrasonidos, las ecuaciones que modelan este tipo de sensores utilizan las expresiones de distancia elegidas referidas a la posición de los sensores $(x_s(t+I), y_s(t+I))$ con respecto al robot $(x(t+I), y(t+I), \theta(t+I))$. La relación existente entre la posición de cada sensor y a la del robot es recogida en las siguientes expresiones, donde d_s es la distancia del sensor al centro del robot y θ_s la orientación del sensor respecto al robot:

$$\begin{aligned} x_s &= x(t+I) + d_s \cos(\theta(t+I) + \theta_s) \\ y_s &= y(t+I) + d_s \sin(\theta(t+I) + \theta_s) \end{aligned} \tag{A5.10}$$

Con todas estas consideraciones, las ecuaciones del modelo no lineal del sensor que hace un uso simultáneo de las medidas de ultrasonidos y un mapa del entorno se recogen, según el tipo de objeto detectado, en (A5-11). La primera ecuación representa la distancia entre el centro del robot y la recta de la pared, la segunda entre el centro del robot y la esquina o rincón, y la tercera entre el

A5.2. Modelos de medida: ecuaciones de los sensores utilizados para determinar la posición del robot

centro del robot y la columna. En el último caso, que se da cuando el ultrasonido no es capaz de predecir ningún objeto del mapa desde la posición que se encuentra, se le asignará un valor constante y elevado, para que las medidas reales que no pueden ser predichas por el modelo, y que pueden ser originadas por un objeto no presente en el mapa o un eco falso, sean directamente desechadas por el test de validación del algoritmo IFAsynMixto.

$$\mathbf{z}_{2+k,t+1} = z_d(t+1) = \begin{cases} |A_{obj}[x(t+1) + d_s \cos(\theta(t+1) + \theta_s)] + B_{obj}[y(t+1) + d_s \sin(\theta(t+1) + \theta_s)] + C_{obj}| & \text{si Obj} = \text{Pared} \\ \sqrt{[x(t+1) + d_s \cos(\theta(t+1) + \theta_s) - x_{obj}]^2 + [y(t+1) + d_s \sin(\theta(t+1) + \theta_s) - y_{obj}]^2} & \text{si Obj} = \begin{cases} \text{Esquina} \\ \text{Rincon} \end{cases} \\ \sqrt{[x(t+1) + d_s \cos(\theta(t+1) + \theta_s) - x_{obj}]^2 + [y(t+1) + d_s \sin(\theta(t+1) + \theta_s) - y_{obj}]^2} - r_{obj} & \text{si Obj} = \text{Columna} \\ \inf & \text{si } \neg \text{Obj} \end{cases} \quad (\text{A5.11})$$

El modelo del sistema es no lineal, y la matriz de medida $\mathbf{H}_{2+k,t+1}$ se obtiene, para cada una de las clases de objetos posibles, como el jacobiano de la expresiones recogidas en (A5-11). Las expresiones resultantes se presentan en (A5-12). En el caso en el que desde la posición estimada del robot el modelo no detecte la presencia de ningún objeto la matriz de medida $\mathbf{H}_{2+k,t+1}$ toma valores nulos. De esta forma, si el algoritmo IFAsynMixto no utiliza el test de validación chi-cuadrado para las medidas asociadas a los ultrasonidos, la información sensorial ($\mathbf{H}_{s,t+1}^T \mathbf{R}_{s,t+1}^{-1} \zeta_{s,t+1}^C$) correspondiente a las medidas que no se asocian a ningún objeto es considerada nula y por lo tanto, no modifica el valor del estado durante la etapa de asimilación.

$$\mathbf{H}_{2+k,t+1} = \mathbf{H}_d(2+k,t+1) = \begin{cases} \begin{pmatrix} (-1)^p A_{obj} & (-1)^p B_{obj} & (-1)^p (B_{obj} d_s \cos(\theta(t+1) + \theta_s) - A_{obj} d_s \sin(\theta(t+1) + \theta_s)) & 0 & 0 \end{pmatrix} & \text{si Obj} = \text{Pared} \\ \begin{pmatrix} \frac{x(t+1) + d_s \cos(\theta(t+1) + \theta_s) - x_{obj}}{q} & \frac{y(t+1) + d_s \sin(\theta(t+1) + \theta_s) - y_{obj}}{q} & w \frac{d_s}{q} & 0 & 0 \end{pmatrix} & \text{si Obj} = \begin{cases} \text{Esquina} \\ \text{Rincon} \\ \text{Columna} \end{cases} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix} & \text{si } \neg \text{Obj} \end{cases} \quad (\text{A5.12})$$

$$\text{con } p = \begin{cases} 0 & \text{si } (A_{obj}[x(t+1) + d_s \cos(\theta(t+1) + \theta_s)] + B_{obj}[y(t+1) + d_s \sin(\theta(t+1) + \theta_s)] + C_{obj}) > 0.0 \\ 1 & \text{c.c.} \end{cases}$$

$$q = \sqrt{[x(t+1) + d_s \cos(\theta(t+1) + \theta_s) - x_{obj}]^2 + [y(t+1) + d_s \sin(\theta(t+1) + \theta_s) - y_{obj}]^2}$$

$$w = (y(t+1) + d_s \sin(\theta(t+1) + \theta_s) - y_{obj}) \cos(\theta(t+1) + \theta_s) - (x(t+1) + d_s \cos(\theta(t+1) + \theta_s) - x_{obj}) \sin(\theta(t+1) + \theta_s)$$

La matriz de covarianza $\mathbf{R}_{2+k,t+1}$ asociada al sensor formado por los ultrasonidos y el mapa se encuentra recogido en la siguiente expresión:

$$\mathbf{R}_{2+k,t+1} = (\text{var}_{sonar_D}) \quad (\text{A5.13})$$

Se ha estudiado una posible alternativa al modelo elegido para nuestros sensores de ultrasonidos que se fundamenta en la elevada directividad de los mismos, y contempla la posibilidad de que

éstos sean utilizados para obtener también información sobre la orientación del robot cuando el objeto detectado sea una pared, utilizando la orientación del objeto como otra componente del vector de medida. De esta forma el vector de medida del nuevo modelo estaría formado por dos componentes, la componente de distancia presentada en las expresiones de (A5-11) y la componente angular, cuyo valor sería únicamente no nulo cuando se detectase una pared. En ese caso la medida proporcionada por el sensor estaría relacionada con los valores de A_{obj} y B_{obj} , ya que éstos definen la orientación del vector normal a la recta que modela la existencia de la pared en el mapa. El nuevo modelo de medida se recoge en la expresión (A5-14), donde se obvia la definición de $z_d(t+1)$ ya que ésta es la misma que la establecida en (A5-11).

$$\mathbf{z}_{2+k,t+1} = \begin{pmatrix} z_d(t+1) \\ z_\theta(t+1) \end{pmatrix} \quad (A5.14)$$

$$con \quad z_\theta(t+1) = \begin{cases} \theta(t+1) + \theta_s & \text{si } Obj = Pared \\ 0 & \text{si } (\neg Obj) \vee (Obj = \neg Pared) \end{cases}$$

La matriz de medida $\mathbf{H}_{2+k,t+1}$ para el nuevo modelo del sensor, presentada en la expresión (A5-15), tendrá una fila adicional con valores nulos salvo para su tercer elemento en el caso en el que se detecte un objeto y éste sea una pared. Este hecho aseguraría la asimilación de información sensorial angular asociada al segundo elemento del vector de medida únicamente cuando el sonar detectase una pared.

$$\mathbf{H}_{2+k,t+1} = \begin{pmatrix} \mathbf{H}_d(2+k,t+1) \\ \mathbf{H}_\theta(2+k,t+1) \end{pmatrix} \quad (A5.15)$$

$$con \quad \mathbf{H}_\theta(2+k,t+1) = \begin{cases} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \end{pmatrix} & \text{si } Obj = Pared \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix} & \text{si } (\neg Obj) \vee (Obj = \neg Pared) \end{cases}$$

La matriz de covarianza $\mathbf{R}_{2+k,t+1}$ asociada al sensor formado por los ultrasonidos y el mapa que proporcionan información angular cuando el objeto detectado es una pared se encuentra recogida en la siguiente expresión:

$$\mathbf{R}_{2+k,t+1} = \begin{pmatrix} \text{var}_{sonar_D} & 0 \\ 0 & \text{var}_{sonar_\theta} \end{pmatrix} \quad (A5.16)$$

Entre los dos modelos planteados se ha implementado el primero ya que el objeto de las pruebas reales es observar el funcionamiento del algoritmo IFAsynMixto dentro de un sistema de control real, y según se eleva la complejidad de los modelos es más difícil analizar el funcionamiento del filtro. La complejidad del segundo modelo propuesto se ve claramente incrementada por la inclusión de una componente adicional, que al ser una medida directamente asociada con la orientación del robot plantea los mismos problemas de asimilación y cálculo de innovación que los planteados en el caso de la brújula. Por lo tanto, en una implementación del segundo modelo sería

necesario incluir operaciones de cálculo de innovación y de corrección de medida especiales para este sensor, que tengan en cuenta el problema de la discontinuidad existente en los valores angulares. Con la elección del primer modelo, que solo proporciona medidas de distancia, ligadas a la posición (x,y) y orientación (θ) del robot, evitamos estos problemas reduciendo la complejidad del modelo del sensor de ultrasonidos con mapa, que se encuentra replicado en el algoritmo IFAsynMixto para los 8 sensores de ultrasonidos del robot. Como contrapartida, esta elección hace que las medidas del sistema ultrasonidos + mapa proporcionen menos información al algoritmo IFAsynMixto.

A5.3. Versión adaptable del algoritmo IFAsynMixto

Las peculiaridades observadas en las medidas de orientación hacen necesario modificar el algoritmo IFAsynMixto para el sistema cuyos modelos se describen en la sección anterior, o desarrollar una versión adaptable del mismo, con el objeto de poder reutilizarla para problemas generales que presentasen problemáticas similares. Se ha optado por la segunda solución, debido a su versatilidad posterior. La versión adaptable del algoritmo IFAsynMixto, implementada únicamente en C++ y que se presenta a continuación, permite utilizar todas las operaciones habituales del algoritmo IFAsynMixto y modificar la funcionalidad de alguna de ellas a través de la definición de operaciones especiales para un modelo particular. De esta forma, la nueva versión del algoritmo es más sencilla de reconfigurar y podrá ser utilizada en distintos sistemas sin tener que realizar modificaciones continuas del algoritmo. La generalización conseguida podrá ser aumentada para otros problemas cuando sea necesario, siguiendo una estrategia similar a la planteada hasta el momento con las modificaciones ya incorporadas.

Las principales modificaciones de la nueva versión, que se presenta a continuación, son las siguientes:

- Se incorporara la posibilidad de definir en el modelo de transición una función, denominada *RangoValido()*, para llevar los valores del vector de estado al rango adecuado, que es ejecutada siempre y cuando el estado pueda ser modificado (fenómeno que ocurre en la etapa de predicción y en la asimilación de la información sensorial). La función definida por defecto no realiza ningún cambio en los valores del estado. En nuestro problema particular esta función es necesaria para asegurar que el estado angular se encuentra siempre en el rango válido de valores $[0, 2\pi)$.
- El usuario también puede especificar para cada uno de los sensores una función, denominada *Innovacion_s()*, responsable del cálculo de la innovación $\hat{\mathbf{r}}_{s,k/AASIM \leq k-1}$ que es utilizada en la etapa de validación. Si la función no es definida para un sensor particular el algoritmo

utilizará la función por defecto, que calcula la diferencia existente entre la medida y el valor predicho:

$$Innovacion_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, \zeta_{s,k}, k) = \zeta_{s,k} - h_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k)$$

En nuestro problema particular esta función es necesaria para el cálculo del valor de la innovación asociada a la brújula, debido a la discontinuidad existente en los valores de tipo angular.

- Cualquier tipo de medida, sea lineal o no lineal, puede ser corregida por el algoritmo antes de ser convertida en información sensorial. Este proceso se realiza a través de una función, denominada *CorregirMedida_s()*, que puede ser definida dentro del modelo de medida de cada sensor para aquellos casos en los que la corrección estándar no sea válida. La operación realizada por defecto, cuando en el modelo del sensor no se ha definido ninguna función especial, depende de si el sensor es lineal, caso en el que no se realiza ninguna modificación, o si no lo es, caso en el que se realiza la corrección habitual de la etapa de asimilación del algoritmo de información extendido.

$$CorregirMedida_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, \zeta_{s,k}, k) = \begin{cases} \zeta_{s,k} & \text{si } SLineal(s)=true \\ \zeta_{s,k} - h_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k) + ObtenerH_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k) \cdot \hat{\mathbf{x}}_{t/AASIM \leq k-1} & \text{c.c.} \end{cases}$$

En nuestro problema particular esta función es necesaria para el cálculo del valor de la medida modificada asociada a la brújula, debido a la discontinuidad existente en los valores de tipo angular.

- La variable *SLineal*, que se utilizaba en la versión original de IFAsynMixto para determinar si un sensor era lineal o no, puede ser utilizada para indicarle al algoritmo que sensores deben de ser utilizados como si se tratasen de sensores no lineales, para que sus valores de medida sean almacenados y su información sensorial recalculada cuando lleguen al algoritmo medidas retardadas. Este uso especial de la variable debe realizarse con cuidado, ya que aunque siempre es posible dar un tratamiento no lineal a un sensor lineal, no es conveniente incrementar las necesidades de memoria y de cómputo del algoritmo sin causa justificada. Para nuestro problema, se ha definido la brújula como un sensor que debe ser tratado de forma no lineal a pesar de la linealidad por tramos de las ecuaciones de su modelo, debido a la discontinuidad existente en los valores angulares.

Con estas modificaciones, la versión adaptable del algoritmo IFAsynMixto es recogida en el siguiente algoritmo.

- 1) Inicialización
 - a. Obtener el modelo del sistema (ecuación de transición de estados) y de los sensores (ecuaciones de medida), el vector **SLineal** que determinará que sensores **s** deben recibir un tratamiento lineal (**SLineal(s)=true**) y cuales no (**SLineal(s)=false**), y la matriz **SAsimilada**, inicializada a **false**, que determinará si el algoritmo ha asimilado una medida para el sensor **s** en el instante **k** (**SAsimilada(k,s)=true**) o no **SAsimilada(k,s)=false**.
 - b. Fijar el tamaño de la ventana de tiempo **n**.
 - c. **t=1**
 - d. Inicializar las variables asociadas al instante **t=1**

$$\mathbf{F}_0 = \text{ObtenerF}(\hat{\mathbf{x}}_0, \mathbf{u}_0, 0)$$

$$\mathbf{Y}_{1/AASIM \leq 0} = (\mathbf{F}_0 \mathbf{P}_0 \mathbf{F}_0^T + \mathbf{Q}_0)^{-1}$$

$$\hat{\mathbf{y}}_{1/AASIM \leq 0} = \mathbf{Y}_{1/AASIM \leq 0} \cdot \text{RangoValido}(f(\hat{\mathbf{x}}_0, \mathbf{u}_0, 0))$$

$$\mathbf{iL}_1 = \mathbf{0}, \mathbf{IL}_1 = \mathbf{0}, \mathbf{iNL}_1 = \mathbf{0}, \mathbf{INL}_1 = \mathbf{0}$$
- 2) Bucle principal: **while(true)**
 - a. Si ha pasado el tiempo de un muestreo, hay que inicializar las variables del nuevo instante de muestreo, y posiblemente eliminar las variables del periodo más antiguo:
 - i. **t=t+1**
 - ii. Se asimila la información del instante anterior t-1
$$\hat{\mathbf{y}}_{t-1/AASIM \leq t-1} = \hat{\mathbf{y}}_{t-1/AASIM \leq t-2} + \mathbf{iL}_{t-1} + \mathbf{iNL}_{t-1}, \quad \mathbf{Y}_{t-1/AASIM \leq t-1} = \mathbf{Y}_{t-1/AASIM \leq t-2} + \mathbf{IL}_{t-1} + \mathbf{INL}_{t-1}$$
 - iii. Se inicializan las variables para el instante t
$$\hat{\mathbf{x}}_{t-1/AASIM \leq t-1} = \mathbf{Y}_{t-1/AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t-1/AASIM \leq t-1}, \quad \mathbf{F}_{t-1} = \text{ObtenerF}(\hat{\mathbf{x}}_{t-1/AASIM \leq t-1}, \mathbf{u}_{t-1}, t-1)$$

$$\mathbf{Y}_{t/AASIM \leq t-1} = (\mathbf{F}_{t-1} \mathbf{Y}_{t-1/AASIM \leq t-1}^{-1} \mathbf{F}_{t-1}^T + \mathbf{Q}_{t-1})^{-1}$$

$$\hat{\mathbf{y}}_{t/AASIM \leq t-1} = \mathbf{Y}_{t/AASIM \leq t-1} \cdot \text{RangoValido}(f(\hat{\mathbf{x}}_{t-1/AASIM \leq t-1}, \mathbf{u}_{t-1}, t-1))$$

$$\mathbf{iL}_t = \mathbf{0}, \mathbf{IL}_t = \mathbf{0}, \mathbf{iNL}_t = \mathbf{0}, \mathbf{INL}_t = \mathbf{0}$$
 - iv. Si **t-n>0** hay que eliminar los elementos asociados al instante **t-n**, es decir las variables:
$$\mathbf{Y}_{t-n/AASIM \leq t-n-1}, \hat{\mathbf{y}}_{t-n/AASIM \leq t-n-1}, \mathbf{u}_{t-n}, \mathbf{iL}_{t-n}, \mathbf{IL}_{t-n}, \mathbf{iNL}_{t-n}, \mathbf{INL}_{t-n}$$

$$\zeta_{s,t-n} \text{ para los } \mathbf{s} \text{ no lineales}$$
 - b. Si hay una medida $\zeta_{s,k}$ disponible, hay que comprobar si ésta se ha realizado en un **k** que esta dentro de la ventana temporal **n** elegida.
 - i. Si no (**k>t-n**) la medida no se puede asimilar, por lo que es conveniente mandar un mensaje al usuario para que detecte que con el **n** elegido se pierden medidas
 - ii. Si (**k>t-n**) la medida se podrá asimilar, siempre que sea válida:
 1. Obtener la covarianza del valor estimado de la medida
$$\hat{\mathbf{x}}_{k/AASIM \leq k-1} = \mathbf{Y}_{k/AASIM \leq k-1}^{-1} \hat{\mathbf{y}}_{k/AASIM \leq k-1}, \quad \mathbf{H}_{s,k} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, k)$$

$$\mathbf{S}_{s,k/AASIM \leq k-1} = \mathbf{H}_{s,k} \mathbf{Y}_{k/AASIM \leq k-1}^{-1} \mathbf{H}_{s,k}^T + \mathbf{R}_{s,k}$$
 2. Realizar el test chi-cuadrado de validación basado en el valor estimado de la medida
$$\hat{\mathbf{r}}_{s,k/AASIM \leq k-1} = \text{Innovacion}_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, \zeta_{s,k}, k)$$

$$\text{Test: } \hat{\mathbf{r}}_{s,k/AASIM \leq k-1}^T \mathbf{S}_{s,k/AASIM \leq k-1}^{-1} \hat{\mathbf{r}}_{s,k/AASIM \leq k-1} \leq \gamma_s \quad ?$$

```

3. Si la medida es válida:
a. Obtener la información sensorial asociada a la medida (en b.ii.1 ya
se ha obtenido la  $\mathbf{H}_{s,k}$  y  $\hat{\mathbf{x}}_{k/AASIM \leq k-1}$ ) e indicar que va a ser asimilada
 $\zeta_{s,k}^C = \text{CorregirMedida}_s(\hat{\mathbf{x}}_{k/AASIM \leq k-1}, \zeta_{s,k}, k)$ 
 $\mathbf{i}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \zeta_{s,k}^C$  ,  $\mathbf{I}_{s,k} = \mathbf{H}_{s,k}^T \mathbf{R}_{s,k}^{-1} \mathbf{H}_{s,k}$ 
if SLineal(s) = false , Almacenar  $\zeta_{s,k}$  , end
SAsimilada(k, s) = true

b. Añadir la nueva información sensorial al resto de la información
sensorial, del mismo tipo, asociada a dicho periodo de muestreo:
if SLineal(s) = 1  $\mathbf{iL}_k = \mathbf{iL}_k + \mathbf{i}_{s,k}$  ,  $\mathbf{IL}_k = \mathbf{IL}_k + \mathbf{I}_{s,k}$ 
else  $\mathbf{iNL}_k = \mathbf{iNL}_k + \mathbf{i}_{s,k}$  ,  $\mathbf{INL}_k = \mathbf{INL}_k + \mathbf{I}_{s,k}$ 
end

c. Si  $k \neq t$ , hay que propagar la nueva información sensorial hacia
delante, de forma que se actualicen el término aditivo de la etapa de
asimilación que incluye la información sobre el estado en instante de
muestreo anteriores:
Se asimila la información del instante k y se realiza la 1ª
predicción
 $\hat{\mathbf{y}}_{k/AASIM \leq k} = \hat{\mathbf{y}}_{k/AASIM \leq k-1} + \mathbf{iL}_k + \mathbf{iNL}_k$  ,  $\mathbf{Y}_{k/AASIM \leq k} = \mathbf{Y}_{k/AASIM \leq k-1} + \mathbf{IL}_k + \mathbf{INL}_k$ 
 $\hat{\mathbf{x}}_{k/AASIM \leq k} = \text{RangoValido}(\mathbf{Y}_{k/AASIM \leq k}^{-1} \hat{\mathbf{y}}_{k/AASIM \leq k})$   $\mathbf{F}_k = \text{ObtenerF}(\hat{\mathbf{x}}_{k/AASIM \leq k}, \mathbf{u}_k, k)$ 
 $\mathbf{Y}_{k+1/AASIM \leq k} = (\mathbf{F}_k \mathbf{Y}_{k/AASIM \leq k}^{-1} \mathbf{F}_k^T + \mathbf{Q}_k)^{-1}$ 
 $\hat{\mathbf{y}}_{k+1/AASIM \leq k} = \mathbf{Y}_{k+1/AASIM \leq k} \cdot \text{RangoValido}(f(\hat{\mathbf{x}}_{k/AASIM \leq k}, \mathbf{u}_k, k))$ 
for  $j = k+1:t-1$ 
Se recalcula la información sensorial no lineal
 $\hat{\mathbf{x}}_{j/AASIM \leq j-1} = \mathbf{Y}_{j/AASIM \leq j-1}^{-1} \hat{\mathbf{y}}_{j/AASIM \leq j-1}$  ,  $\mathbf{iNL}_j = \mathbf{0}$  ,  $\mathbf{INL}_j = \mathbf{0}$ 
for  $s = 1:S$ 
if SLineal(s) = false  $\wedge$  SAsimilada(j, s) = true
 $\mathbf{H}_{s,j} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{j/AASIM \leq j-1}, j)$ 
 $\zeta_{s,j}^C = \text{CorregirMedida}_s(\hat{\mathbf{x}}_{j/AASIM \leq j-1}, \zeta_{s,j}, j)$ 
 $\mathbf{i}_{s,j} = \mathbf{H}_{s,j}^T \mathbf{R}_{s,j}^{-1} \zeta_{s,j}^C$  ,  $\mathbf{I}_{s,j} = \mathbf{H}_{s,j}^T \mathbf{R}_{s,j}^{-1} \mathbf{H}_{s,j}$ 
 $\mathbf{iNL}_j = \mathbf{iNL}_j + \mathbf{i}_{s,j}$  ,  $\mathbf{INL}_j = \mathbf{INL}_j + \mathbf{I}_{s,j}$ 
end
end
Se asimila la información sensorial para el instante j y se
realiza la siguiente predicción
 $\hat{\mathbf{y}}_{j/AASIM \leq j} = \hat{\mathbf{y}}_{j/AASIM \leq j-1} + \mathbf{iL}_j + \mathbf{iNL}_j$ 
 $\mathbf{Y}_{j/AASIM \leq j} = \mathbf{Y}_{j/AASIM \leq j-1} + \mathbf{IL}_j + \mathbf{INL}_j$ 
 $\hat{\mathbf{x}}_{j/AASIM \leq j} = \text{RangoValido}(\mathbf{Y}_{j/AASIM \leq j}^{-1} \hat{\mathbf{y}}_{j/AASIM \leq j})$ 
 $\mathbf{F}_j = \text{ObtenerF}(\hat{\mathbf{x}}_{j/AASIM \leq j}, \mathbf{u}_j, j)$ 
 $\mathbf{Y}_{j+1/AASIM \leq j} = (\mathbf{F}_j \mathbf{Y}_{j/AASIM \leq j}^{-1} \mathbf{F}_j^T + \mathbf{Q}_j)^{-1}$ 
 $\hat{\mathbf{y}}_{j+1/AASIM \leq j} = \mathbf{Y}_{j+1/AASIM \leq j} \cdot \text{RangoValido}(f(\hat{\mathbf{x}}_{j/AASIM \leq j}, \mathbf{u}_j, j))$ 
end
Se recalcula la información sensorial no lineal del último instante
de muestreo

```

```

 $\hat{\mathbf{x}}_{t|AASIM \leq t-1} = \mathbf{Y}_{t|AASIM \leq t-1}^{-1} \hat{\mathbf{y}}_{t|AASIM \leq t-1} \quad , \quad \mathbf{iNL}_t = \mathbf{0} \quad , \quad \mathbf{INL}_t = \mathbf{0}$ 
for  $s = 1 : S$ 
    if  $SLineal(s) = \text{false} \wedge SAsimilda(t, s) = \text{true}$ 
         $\mathbf{H}_{s,t} = \text{ObtenerH}_s(\hat{\mathbf{x}}_{t|AASIM \leq t-1}, t)$ 
         $\zeta_{s,t}^C = \text{CorregirMedida}_s(\hat{\mathbf{x}}_{t|AASIM \leq t-1}, t, \zeta_{s,t}, t)$ 
         $\mathbf{i}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \zeta_{s,t}^C \quad , \quad \mathbf{I}_{s,t} = \mathbf{H}_{s,t}^T \mathbf{R}_{s,t}^{-1} \mathbf{H}_{s,t}$ 
         $\mathbf{iNL}_t = \mathbf{iNL}_t + \mathbf{i}_{s,t} \quad , \quad \mathbf{INL}_t = \mathbf{INL}_t + \mathbf{I}_{s,t}$ 
    end
end
end

```

Algoritmo A5-1. Filtro de Información Mixto reconfigurable para un sistema de control con múltiples sensores asíncronos cuyas medidas pueden estar disponibles de forma temporalmente desordenada, y cuya función de transición puede ser lineal o no lineal, y cada una de sus funciones de medida lineales o no lineales.

